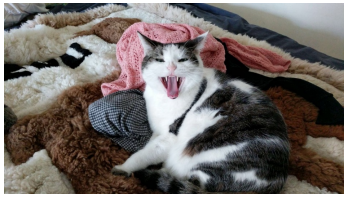
Inference on simple linear regression and regression diagnostics





Overview

Review of inference for simple linear regression

Regression diagnostics

A grammar for data wrangling

Grammar: here we mean a set of components that can be put together to achieve a goal

dplyr is a package in the tidyverse that has a set of verbs that are useful for wrangling data:

1. filter()
2. select()
3. mutate()
4. arrange()
5. group_by()
6. summarize()

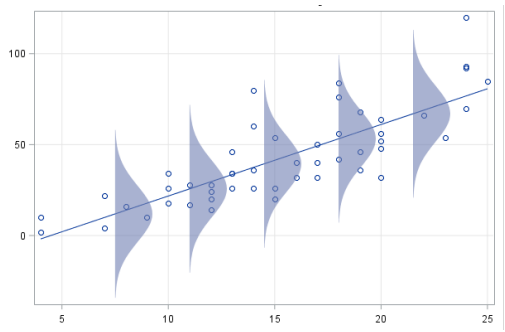
All these function **take a data frame** and other arguments and **return a data**

```
> library(dplyr)   # load the dplyr package
```

Simple linear regression underlying model

Intercept Slope } *Parameters*

$$Y \approx \beta_0 + \beta_1 x \qquad Y = \beta_0 + \beta_1 x + \epsilon$$



$$\epsilon \sim N(0, \sigma_\epsilon)$$

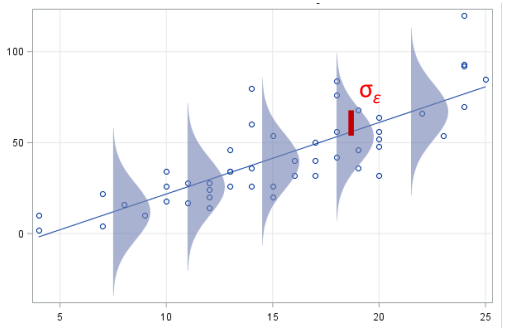
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 + \hat{\beta}_1 x_i)^2$$

Estimating σ_ε

We can also use the **standard deviation of errors** as an estimate standard deviation of irreducible noise $\boldsymbol{\sigma_\varepsilon}$

- This is also called the **residual standard error (RSE)**



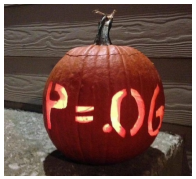
$$\hat{\sigma}_\epsilon = \sqrt{\frac{1}{n-2}SSE}$$

$$= \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Hypothesis test and confidence intervals for regression coefficients

We can run hypothesis tests to assess whether there is a relationship between y and x, and calculate p-values

- H_0: $\beta_1 = 0$ (slope is 0, so no relationship between x and y
- H_A: $\beta_1 \neq 0$



Both the t-test and an ANOVA give the same results for this test

- $F = t^2$

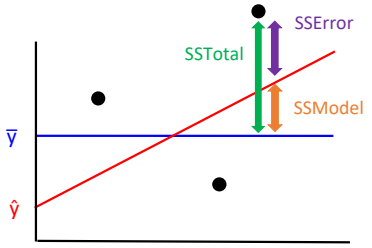
A confidence interval for the true regression slope is given by:

$$\hat{\beta}_1 \pm t^*_{n-2} \cdot SE_{\hat{\beta}_1} \quad \text{where} \quad SE_{\hat{\beta}_1} = \sigma_\epsilon \sqrt{\frac{1}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2}}$$

Hypothesis test based on ANOVA for regression

The ANOVA decomposes the variance as:

- SSTotal = SSModel + SSError

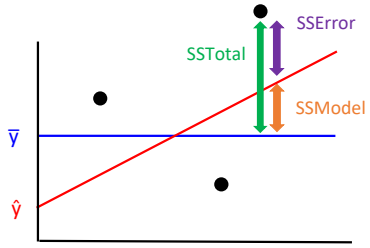


$$\Sigma_{i=1}^{n}(y_i - \bar{y})^2 = \Sigma_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Hypothesis test based on ANOVA for regression

The **percentage of the total variability explained by the model** is given by

$$r^2 = \frac{SSModel}{SSTotal} = 1 - \frac{SSError}{SSTotal}$$

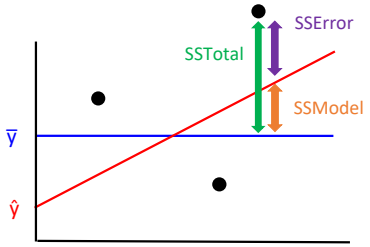


Hypothesis test based on ANOVA for regression

$$F = \frac{SSModel/df_{model}}{SSError/df_{error}} \qquad df_{model} = 1 \qquad df_{error} = n - 2$$

If the null hypothesis is true that $\beta_0 = 0$:

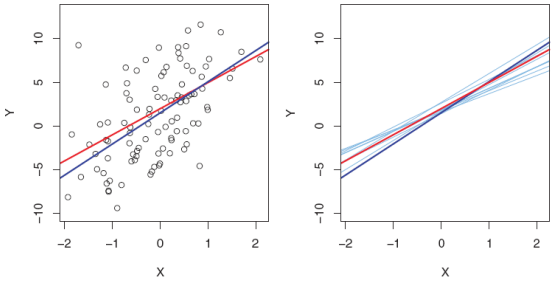
- Both the numerator and denominator are estimates of σ^2
- F comes from an F-distribution with df_{model}, df_{error} degrees of freedom



Resampling methods for inference in regression

We can also use resampling methods to estimate run hypothesis tests and create confidence intervals for the regression coefficients

- Bootstrap
- Permutation test



Let's go back to looking at inference for simple linear regression in R

Using faculty salaries…

Homework 7

edmunds.com℠

price_bought ≈ $\beta_0 + \beta_1$ x mileage_bought + ε

Regression diagnostics

Recall: When using parametric methods, we usually make the following assumptions:

- **Normality**: residuals are normally distributed around the predicted value ŷ
- **Homoscedasticity**: constant variance over the whole range of x values
- **Linearity**: A line can describe the relationship between x and y
- **Independence**: each data point is independent from the other points

We often check these assumptions using regression diagnostic plots

Regression diagnostics

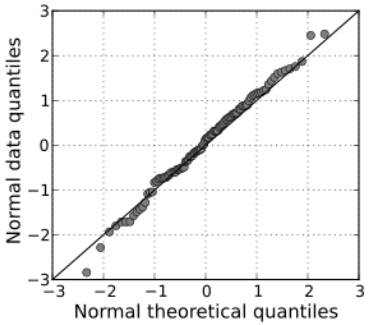
When the regression diagnostics show our assumptions aren't met (or our model is not a good fit) we often adjust the model and try again

Checking normality

Normality: residuals are normally distributed around the predicted value ŷ

We can check this using a Q-Q plot

The 'car' package has a nice function for making qqplots called qqPlot()



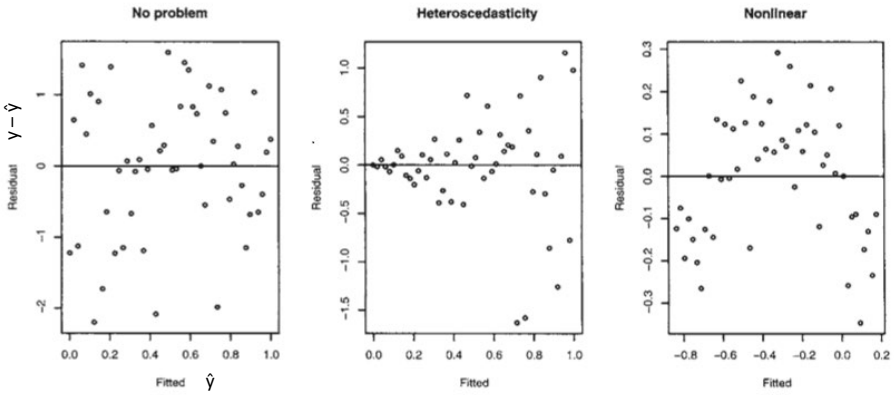
Checking homoscedasticity and linearity

Homoscedasticity: constant variance over the whole range of x values

Linearity: A line can describe the relationship between x and y

We can check homoscedasticity and linearity by plotting the residuals as a function of the fitted values

Checking homoscedasticity and linearity



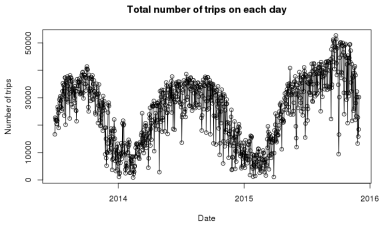
Checking independence

To check whether each data point is independent requires knowledge of how the data was collected

- Simple random sample from the population is likely independent
- Time series of weather not independent

We have basically been assuming independence for everything we have done in this class

- i.i.d. independent and identically distributed



Let's examine these diagnostic plots in R…

Regression diagnostics continued and multiple regression

Overview

Regression diagnostics continued

Confidence and prediction intervals for regression lines

Leverage and influential points

Multiple regression

How did homework 7 go?

edmunds.com SM

price_bought ≈ $\beta_0 + \beta_1$ x mileage_bought + ε

Regression diagnostics

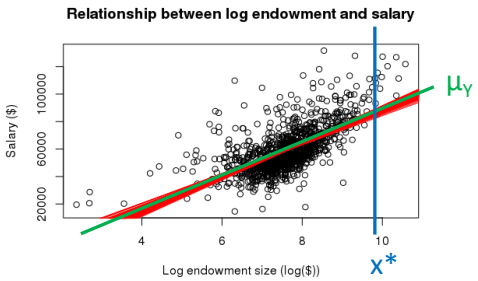
When the regression diagnostics show our assumptions aren't met (or our model is not a good fit) we often adjust the model and try again

Confidence intervals and prediction intervals

Confidence intervals for the regression line μ_Y

A confidence interval for the mean response for the ***true regression line*** μ_Y when X = x* is:

$$\hat{y} \pm t^* \cdot SE_{\hat{\mu}} \quad \text{where} \quad SE_{\hat{\mu}} = \sigma_\epsilon \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2}}$$



Note:

- There is more uncertainty at the ends of the regression line
- The confidence interval for the regression line μ_Y is different than the confidence interval for slope β_1

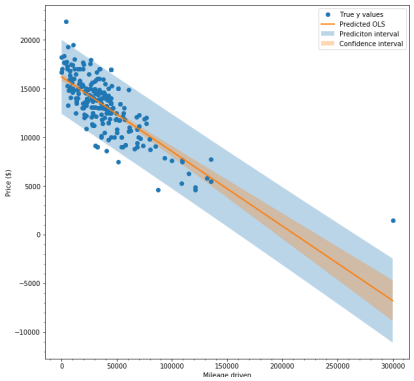
Prediction intervals

Confidence intervals give us a measure of uncertain about our the true relationship between x and y for:

- The true regression slope β_1
- The true regression line μ_Y

Prediction intervals give us a range of plausible values for y

- i.e., 95% of our y's with be within this range



Prediction intervals

A **prediction intervals** for the y can be calculated using:

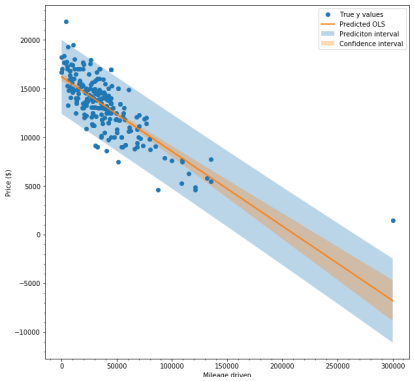
$$\hat{y} \pm t^* \cdot SE_{\hat{y}}$$

where

$$SE_{\hat{y}} = \sigma_\epsilon \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2}}$$

Due to y's scattering around the true regression line

Due to uncertainty in where the true regression line is



Summary of confidence and prediction intervals

1. CI for Slope β $\quad \hat{\beta}_1 \pm t^* \cdot SE_{\hat{\beta}_1} \qquad SE_{\hat{\beta}_1} = \sigma_\epsilon \sqrt{\frac{1}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2}}$

2. CI for regression line μ_Y at point x*

$$\hat{y} \pm t^* \cdot SE_{\hat{\mu}} \qquad SE_{\hat{\mu}} = \sigma_\epsilon \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2}}$$

3. Prediction interval y

$$\hat{y} \pm t^* \cdot SE_{\hat{y}} \qquad SE_{\hat{y}} = \sigma_\epsilon \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2}}$$

Leverage

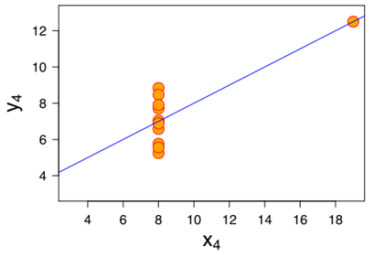
Predictors **x** that are far from the mean have the potential to greatly influence the regression line

- This is known as **leverage**

We can calculate the leverage a data point has using the statistic:

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\Sigma_{j=1}^{2}(x_j - \bar{x})^2}$$

High leverage points can have a big impact on the model that is fit!!!



$$\Sigma_{i=1}^{n} h_i = 2$$

Typical: $h_i = 2/n$
High: $h_i = 4/n$
Very high: $h_i = 6/n$

Standardized and residuals

The **standardized residual** for the i[th] data point in a regression model can be computed using:

$$stdres_i = \frac{y_i - \hat{y}}{\hat{\sigma}_\epsilon \sqrt{1-h_i}}$$

Puts residuals on a 'normalized' scale

Makes residuals at the ends a bit larger to deal with the fact that they are 'overfit'

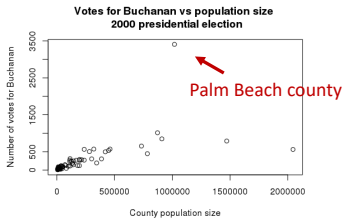
R: rstandard()

Studentized residuals

The **studentized residual** for the i[th] data point in a regression model can be computed using:

$$studres_i = \frac{y_i - \hat{y}}{\hat{\sigma}_{(i)} \sqrt{1-h_i}}$$

Here $\hat{\sigma}_{(i)}$ is the an estimate of $\hat{\sigma}_\epsilon$ with the i[th] point removed



Q: Why might we want to remove the i[th] point when calculating $\hat{\sigma}_\epsilon$?

A: Outliers could have a big effect on our estimate of $\hat{\sigma}_\epsilon$

R: rstudent ()

Cook's distance

The amount of influence a point has on a regression line depends on:

- The size of the residual e_i
- The amount of leverage h_i

Cook's distance is a statistic that captures how much influence a point has on a regression line

$$D_i = \frac{(stdres_i)^2}{k+1} \frac{h_i}{1-h_i}$$

Larger for larger residuals (outliers)

Larger for high leverage points

Where *k* is the number of predictors in the model

- For simple linear regression k = 1 (just a single predictor x)

Cook's distance

The amount of influence a point has on a regression line depends on:

- The size of the residual e_i
- The amount of leverage h_i

Cook's distance is a statistic that captures how much influence a point has on a regression line

$$D_i = \frac{(stdres_i)^2}{k+1} \frac{h_i}{1-h_i}$$

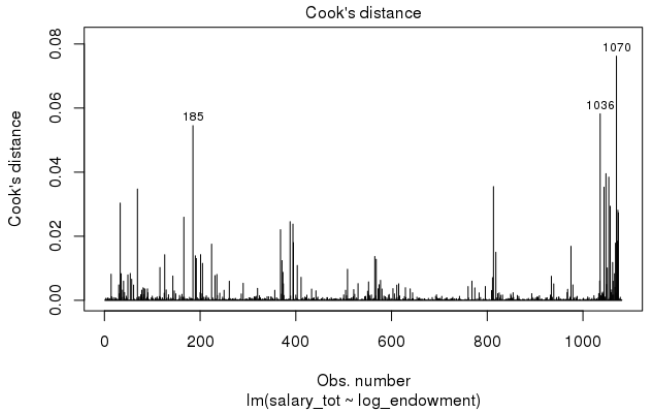
Larger for larger residuals (outliers)

Larger for high leverage points

Rule of thumb:

- Moderately influential: $D_i > 0.5$
- Very influential: $D_i > 1$

Cook's distances for salary ~ $\log_{10}$(endowment)



plot(lm_fit, 4)

Unusual points rules of thumb

Statistic	Moderately unusual	Very unusual
Leverage, h_i	Above 2(k + 1)/n	Above 3(k + 1)/n
Standardized residual	Beyond ± 2	Beyond ± 3
Studentized residual	Beyond ± 2	Beyond ± 3
Cook's D	Above 0.5	Above 1.0

From STAT2 by Cannon, Cobb, Hartlaub, Legler, Lock, Moore, Rossman, and Witmer

Leverage, Cook's distance and other plots in R…

Multiple regression

In multiple regression we try to predict a quantitative response variable *y* using several predictor variables $x_1, x_2, \ldots, x_k$

For multiple linear regression our equation as the form of:

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots \beta_k \cdot x_k + \epsilon$$

We estimate coefficients using a data set to make predictions ŷ

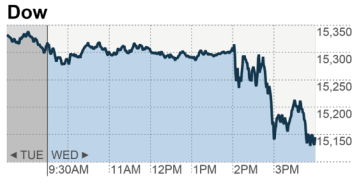
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \ldots + \hat{\beta}_k \cdot x_k$$

Multiple regression

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \ldots + \hat{\beta}_k \cdot x_k$$

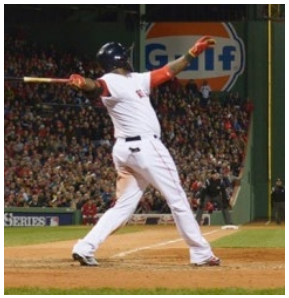
There are many uses for multiple regression models including:

- To make predictions as accurately as possible
- To understand which predictor variables are related to the response variable
- To create new statistics ("metrics") that give a useful numerical description of a phenomenon



Motivation: Who is a better hitter: Derek Jeter or David Ortiz?

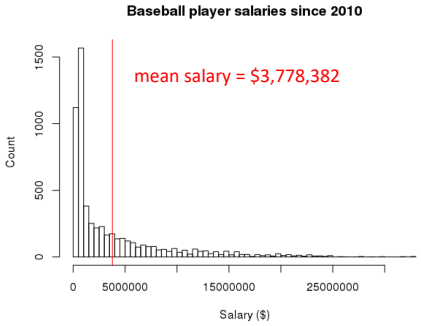
Derek Jeter



David Ortiz

If we are going to pay these players millions of dollars, how can we assess who is best?

Baseball



History of baseball statistics

Henry Chadwick (1824-1908) created the first box score in the 1859 issue of Clipper.

BOSTON.

	T.	R.	1B.	PO.	A.	E.
G. Wright, s. s	6	4	4	1	5	2
Leonard, 2 b.	6	3	3	4	4	3
O'Rourke, 1 b	6	2	3	9	0	1
Murnan, l. f.	6	1	0	3	1	0
Schafer, 2d b	6	3	3	3	1	2
McGinley, c. f	6	0	0	0	0	1
Manning, r. f.	6	0	2	2	0	0
Morrill, c.	6	2	2	4	1	2
Josephs, p.	5	4	4	1	1	2
Totals	53	19	21	27	13	13

ATHLETIC.

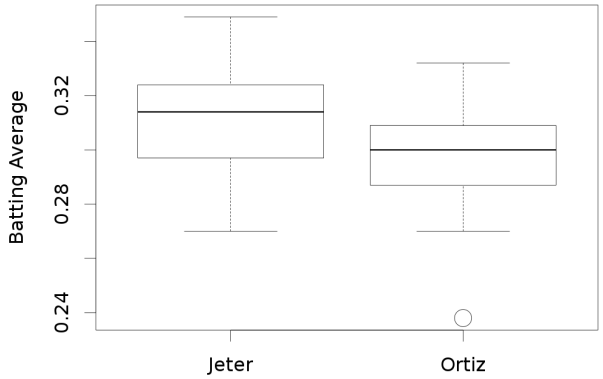
	T.	R.	1B.	PO.	A.	E.
Force, s. s.	5	1	2	1	3	2
Eggler, c. f.	5	3	3	0	0	0
Fisler, r. f.	5	0	1	2	0	0
Meyerle, 3 b	5	1	2	2	3	3
Sutton, 1st b.	5	1	2	10	0	0
Coons, c.	5	1	0	1	1	3
Hall, l. f.	5	1	3	5	0	0
Fowser, 2d b.	6	1	2	6	7	5
Knight, p.	5	2	2	0	1	2
Totals	46	11	17	27	15	15

Boston	0	1	3	3	4	1	0	2	5—19
Athletic	1	0	0	0	3	3	2	2	0—11

Runs earned—Boston, 4; Athletic, 5. Home-run—Hall, 1. Total bases on hits—Boson, 22; Athletic, 20. First base by errors—Boston, 8; Athletic, 5. Umpire, George White of Lowell, Mass. Time 2h. 47m.

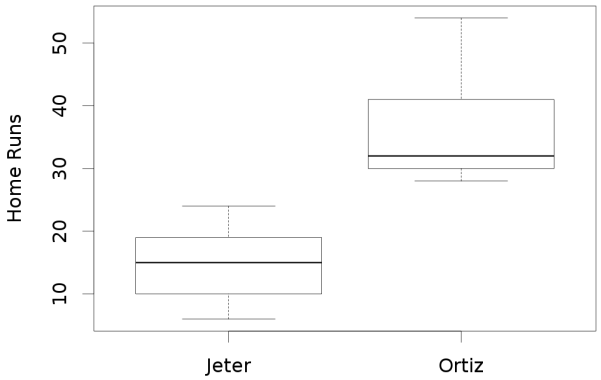
People could now decide who is best player by comparing their statistics!

Who is a better hitter: Derek Jeter or David Ortiz?



Jeter has a better batting average

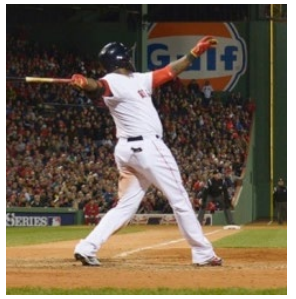
Who is a better hitter: Derek Jeter or David Ortiz?



Ortiz hits more home runs

Who is a better hitter: Derek Jeter or David Ortiz?

Derek Jeter



David Ortiz

How can we decide on the most important descriptive statistic?

Sabermetrics

Sabermetrics: the empirical or mathematical/Statistical study of baseball

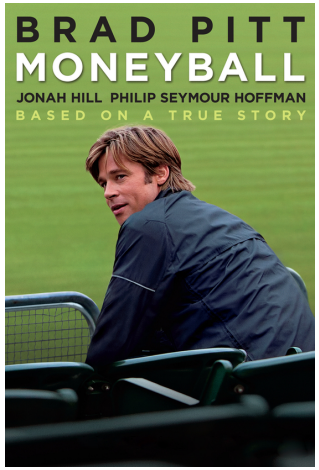
- 'Society for American Baseball Research' (SABR)

Started in the 1970's by Bill James to find more useful measures than classical statistics

- Pre-computers, had to compile all information from old box scores by hand

Billy Bean, the general manager of the A's, used these techniques to create a top ranked team in 2002

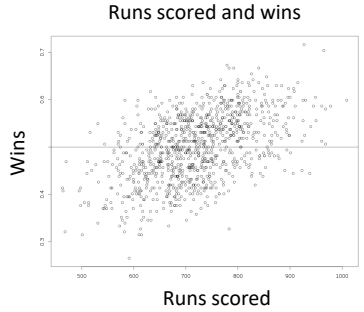
- The book moneyball had a big impact on the expansion of major league clubs doing advanced data analyses



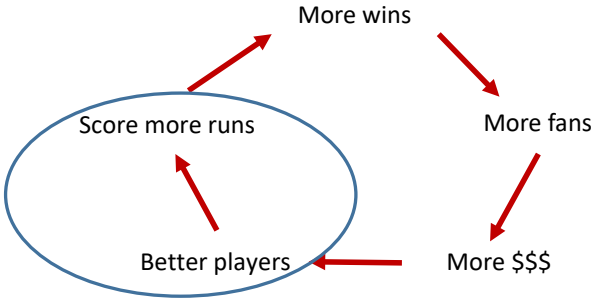
Is power or batting average more important?

It would be good to compare Jeter and Ortiz based on the "best" statistic

How do we determine which statistic is best?

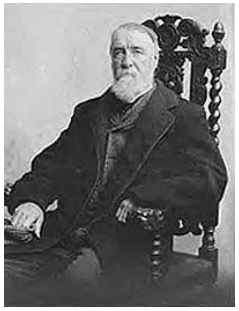


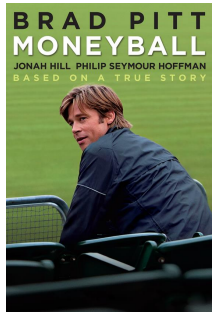
The great cycle of baseball



We can evaluate how 'good' a statistic is based on how well it correlates with the number of runs a team scores

Multiple regression





Overview

Questions

Multiple regression continued

Questions?

Multiple regression

In multiple regression we try to predict a quantitative response variable *y* using several predictor variables $x_1, x_2, \ldots, x_k$

For multiple linear regression our equation as the form of:

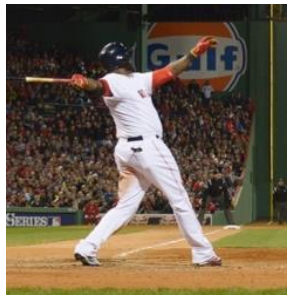
$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots \beta_k \cdot x_k + \epsilon$$

We estimate coefficients using a data set to make predictions ŷ

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \ldots + \hat{\beta}_k \cdot x_k$$

Who is a better hitter: Derek Jeter or David Ortiz?

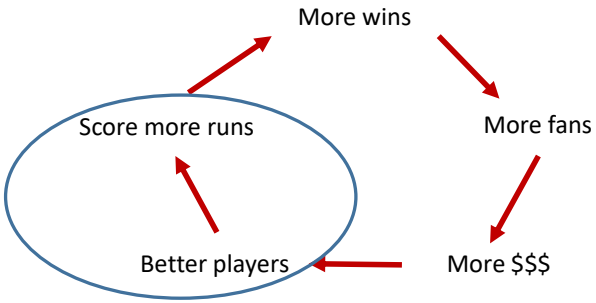
Derek Jeter



David Ortiz

How can we decide on the most important descriptive statistic?

The great cycle of baseball



We can evaluate how 'good' a statistic is based on how well it correlates with the number of runs a team scores

What is the best statistic to use?

<u>One idea</u>: the 'best' statistic to judge a player is the statistic that is most correlated with runs

- We can then use this to examine how good a hitter is

Common baseball descriptive statistics are:

H:	Hits: 1B + 2B + 3B + HR
BB:	Walks: 4 balls
PA:	Plate Appearances: Number of times "up"
AB:	At Bats: PA - BB
OBP:	On-Base Percentage: (H + BB)/PA
BA:	Batting Average: H/AB
SlugPct:	Slugging percentage: (1·1B + 2·2B + 3·3B + 4·HR)/AB

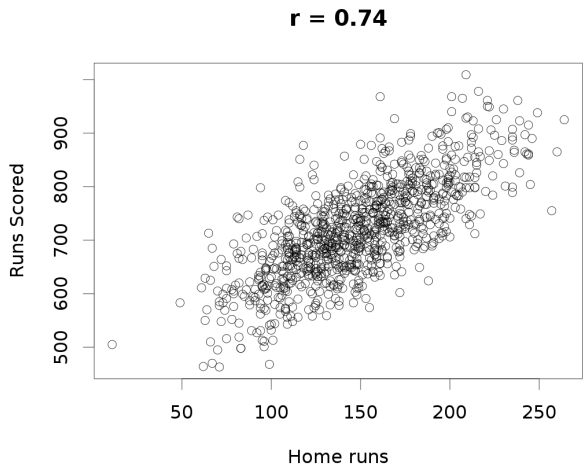
Data exploration in R…

Let's see which statistic is has the highest correlation with how many runs a team scored

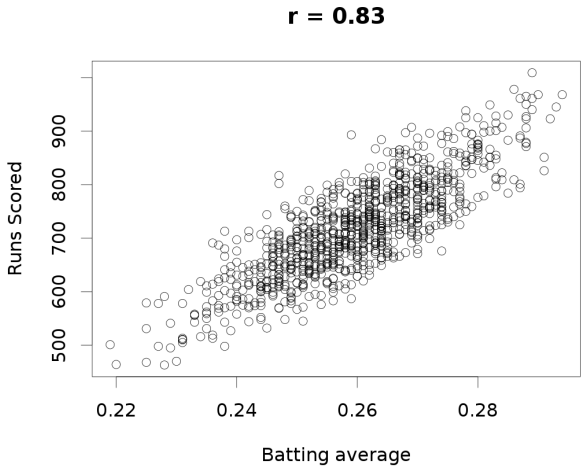
- Using team level data from all baseball seasons with 162 games (i.e., data since 1961)

Data is available in the Lahman package

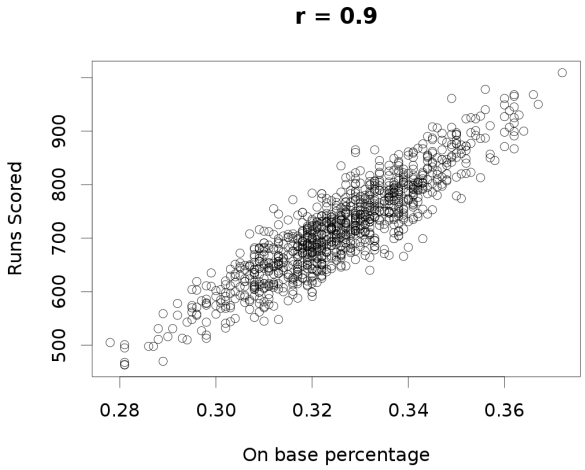
Correlation between HR and runs



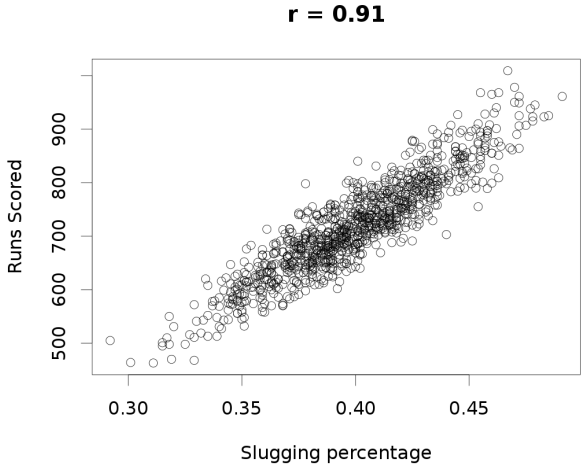
Correlation between BA and runs



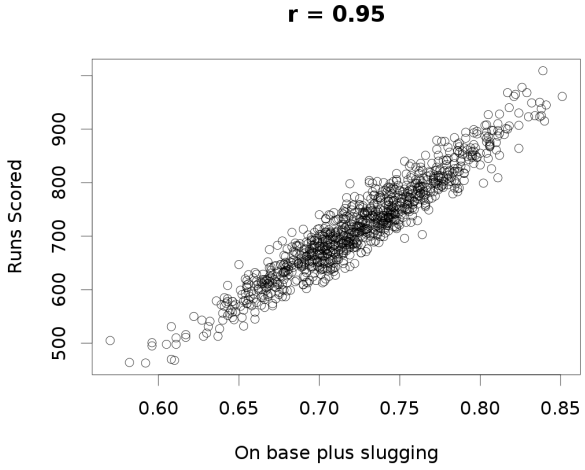
Correlation between OBP and runs



Correlation between Slug and runs



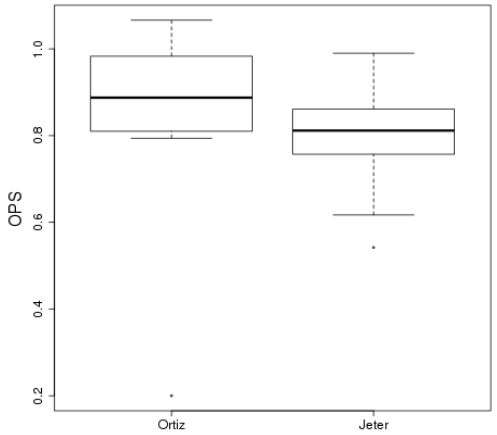
Correlation between OPS and runs



What is the best statistic to use?

It seems like the winner is OPS!

Who is a better hitter: Derek Jeter or David Ortiz?



Ortiz has a better on-base plus slugging!

Better know a player: Derek Jeter

[Onion infographic]

[Other Onion articles]

Creating better 'metrics'

Slugging percentage seemed like the best statistics for predicting runs scored we have found so far

But who says we can't do better!

Creating better 'metrics'

Batting average:

BA = [(1) · **1B** + (1) · **2B** + (1) · **3B** + (1) · **HR**]/**AB**

Slugging percentage:

Slug = [(1) · **1B** + (2) · **2B** + (3) · **3B** + (4) · **HR**]/**AB**

On-base percentage:

OBP = [(1) · **BB** + (1) · **HBP** + (1) · **1B** + (1) · **2B** + (1) · **3B** + (1) · **HR**]/**PA**

Optimal statistic:

OPT = b_1 · **BB** + b_2 · **HBP** + b_3 · **1B** + b_4 · **2B** + b_5 · **3B** + b_6 · **HR** + b_7 · **AB** + b_0

We want to find the "best" b_i's for predicting how many runs a team scored

What are the optimal weights?

Any ideas for the best b_i's ?

OPT = b_1 · BB + b_2 · HBP + b_3 · 1B + b_4 · 2B + b_5 · 3B + b_6 · HR + b_0

Let's use multiple regression to find the b_i's that minimize sum of $(R - OPT)^2$

Let's try it in R…

What are the optimal weights?

Any ideas for the best b_i's ?

OPT = b_1 · BB + b_2 · HBP + b_3 · 1B + b_4 · 2B + b_5 · 3B + b_6 · HR + b_0

Let's use multiple regression to find the b_i's that minimize sum of $(R - OPT)^2$

```
> fit <- lm(R ~ BB + HBP + H + X2B + X3B + HR, data = team_batting)
> coef(fit)
```

What are the optimal weights?

	b_i
(Intercept)	-497.44
HBP	0.42
BB	0.34
X1B	0.56
X2B	0.75
X3B	1.40
HR	1.44

```
> coef(fit)
```

Do these coefficients make sense?

Can you write this in the form of an equation?

What are the optimal weights?

	b_i
(Intercept)	-497.44
HBP	0.42
BB	0.34
X1B	0.56
X2B	0.75
X3B	1.40
HR	1.44

$\hat{r} = .34 \cdot BB + .42 \cdot HBP + .56 \cdot 1B + .75 \cdot 2B + 1.40 \cdot 3B + 1.44 \cdot HR - 497.44$

Multiple linear regression continued

Overview

Discussion of final projects

Multiple linear regression continued

Final projects!

Create a 5-8 page R Markdown report where you analyze your own data

- A chance to practice everything you've learned in class!

Project is due at the end of reading period: December 11th

Sources for data are listed on Canvas

A final project template describing sections that should be in the project will be put on GitHub repository in the homework directory

- This template will be posted by next class

Project report should include the following sections

1. Introduction
   - What is the problem and why is it interesting
   - Where did you get the data (e.g., URL) and what other analyses have been done with this data?

2. Results
   - a) Visualizations: Create visualizations that given insight into the questions you are interested in
   - b) Analyses: Build linear models, run hypothesis tests, create confidence intervals, etc.

3. Conclusions
   - What did you learn

4. Reflection
   - What went well and what was more difficult
   - Any additional things you tried that you did not end up including in this write-up
   - Approximately how much time you spend working the project

Last class we explored the question: who is a better hitter: Derek Jeter or David Ortiz?

Derek Jeter

David Ortiz

We used linear regression to find the 'optimal' (minimized least squares) statistic:

OPT = $b_1 \cdot$ **BB** + $b_2 \cdot$ **HBP** + $b_3 \cdot$ **1B** + $b_4 \cdot$ **2B** + $b_5 \cdot$ **3B** + $b_6 \cdot$ **HR** + $b_7 \cdot$ **AB** + b_0

Multiple Linear Regression

We can calculate the sum of squared errors (SSE)

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \ldots - \hat{\beta}_1 x_{id})^2$$

We can use linear algebra and multivariate calculate to find the coefficients. The solution is:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

The output for our statistic showed

```
(Intercept)         X1B         X2B         X3B          HR          BB          AB
106.4370957   0.6380842   0.8382352   1.3246222   1.5579544   0.3496647  -0.1267687

Call:
lm(formula = R ~ X1B + X2B + X3B + HR + BB + AB, data = team_batting)

Residuals:
    Min      1Q  Median      3Q     Max
-76.685 -15.609  -0.977  14.913  79.020

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 106.43710   73.32426   1.452    0.147
X1B           0.63808    0.01678  38.020  < 2e-16 ***
X2B           0.83824    0.02437  34.393  < 2e-16 ***
X3B           1.32462    0.07352  18.018  < 2e-16 ***
HR            1.55795    0.02812  55.406  < 2e-16 ***
BB            0.34966    0.01073  32.598  < 2e-16 ***
AB           -0.12677    0.01618  -7.835 1.09e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.45 on 1116 degrees of freedom
Multiple R-squared:  0.9294,    Adjusted R-squared:  0.929
F-statistic:  2449 on 6 and 1116 DF,  p-value: < 2.2e-16
```

Estimated coefficients generally make sense

$$\hat{\beta}_{BB} < \hat{\beta}_{1B} < \hat{\beta}_{2B} < \ldots < \hat{\beta}_{HR}$$

But why is at-bats (AB) negative?

- Shouldn't a team score more runs if they have more chances to hit the ball?

summary(lm_fit)

Multiple Linear Regression

The coefficient $\hat{\beta}_i$ we find for predictor x_i shows how much y is predicted by x_i, given the other predictor values x_1, x_2, …, x_k

In particular, each coefficient value is given by how value of x_i, which can not be predicted by the other x's, is able predict y

- We will explore this in R…

For the case of predicting Runs, when the variables X1B, X2B, etc. are already taken into account, having more at-bats (AB) is now negatively associated with runs

- Because those at-bats didn't produce hits but instead produced outs

Adding more variables for better predictions

Last class we also tried to add more variables to our model to make better predictions

```
team_batting2 <- mutate(team_batting,
                        X1Bn = X1B/AB,
                        X2Bn = X2B/AB,
                        X3Bn = X3B/AB,
                        XHRn = HR/AB,
                        XBBn = BB/AB)
```

```
Call:
lm(formula = R ~ (X1B + X2B + X3B + HR + BB + X1Bn + X2Bn + X3Bn +
    XHRn + XBBn), data = team_batting2)

Residuals:
    Min      1Q  Median      3Q     Max
-77.148 -15.696  -0.917  14.911  75.259

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.904e+02  2.139e+01 -27.600   <2e-16 ***
X1B         -7.791e-02  4.423e-01  -0.176    0.860
X2B          1.754e+00  1.377e+00   1.274    0.203
X3B         -1.013e+00  4.727e+00  -0.214    0.830
HR           6.568e-01  1.578e+00   0.416    0.677
BB           3.431e-01  6.605e-01   0.519    0.604
X1Bn         3.943e+03  2.456e+03   1.605    0.109
X2Bn        -5.085e+03  7.611e+03  -0.668    0.504
X3Bn         1.297e+04  2.618e+04   0.495    0.620
XHRn         4.974e+03  8.730e+03   0.570    0.569
XBBn         3.547e+01  3.650e+03   0.010    0.992
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.53 on 1112 degrees of freedom
Multiple R-squared:  0.9292,    Adjusted R-squared:  0.9286
F-statistic:  1460 on 10 and 1112 DF,  p-value: < 2.2e-16
```

Our old variables are no longer statistically significant either ☹

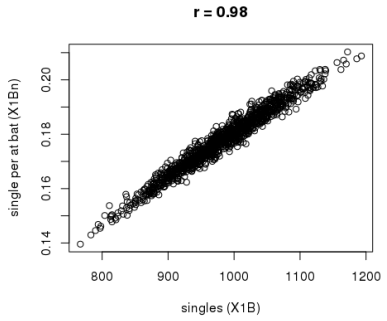
None of our new variables are statistically significant

What is going on?

Multicollinearity

Multicollinearity occurs when two or more variables are closely related to each other

- E.g., if they have a high correlation

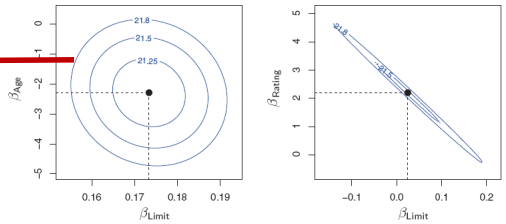


Multicollinearity

Multicollinearity can make our estimate of the regression coefficients unstable

- i.e., a large range of coefficient values can lead to the same standard deviation of errors $\hat{\sigma}_\epsilon$

Contours of equal $\hat{\sigma}_\epsilon$ value



This increases our estimate of the variance of a coefficient and hence can decrease the power to detect a statistically significant predictor

Multicollinearity

The **variance inflated factor** is a statistic that can be computed to test for multicollinearity

$$VIF_i = \frac{1}{1-R_i^2}$$

where R_i^2 is the coefficient of multiple determination for a model to predict x_i using the other predictors in the model

Rule of thumb: suspect multicollinearity for VIF > 5

car::vif(lm_fit)

Are any of the predictors x_i related to y?

We can set this up as a hypothesis test:

H_0: $\beta_1 = \beta_2 = \ldots = \beta_k = 0$

H_A: At least one $\beta_j \neq 0$

To test this we can compute an F-Statistic: $F = \frac{SSModel/k}{SSE/(n-k-1)}$

- Where k is the number of predictors in the model

Under the null hypothesis (and errors that are normally distributed) the F-statistic follows an F-distribution

```
Call:
lm(formula = R ~ (X1B + X2B + X3B + HR + BB + X1Bn + X2Bn + X3Bn +
    XHRn + XBBn + -1), data = team_batting2)

Residuals:
   Min     1Q Median     3Q    Max
-94.28 -20.63  -1.35  18.77 114.34

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
X1B   1.355e+00  5.700e-01   2.378   0.0176 *
X2B   3.170e-01  1.786e+00   0.177   0.8592
X3B   4.598e+00  6.128e+00   0.750   0.4532
HR    4.143e+00  2.041e+00   2.030   0.0426 *
BB   -2.925e-01  8.564e-01  -0.342   0.7328
X1Bn -6.359e+03  3.150e+03  -2.019   0.0437 *
X2Bn  9.474e+02  9.871e+03   0.096   0.9236
X3Bn -2.001e+04  3.393e+04  -0.590   0.5555
XHRn -1.683e+04  1.128e+04  -1.492   0.1359
XBBn  3.600e+03  4.732e+03   0.761   0.4469
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 30.53 on 1113 degrees of freedom
Multiple R-squared:  0.9982,    Adjusted R-squared:  0.9982
F-statistic: 6.29e+04 on 10 and 1113 DF,  p-value: < 2.2e-16
```

Only a few coefficients are significant at the α = 0.05 level

Overall H_0: $\beta_1 = \beta_2 = \ldots = \beta_k = 0$ is highly significant

This can happen when there is multicolinearity

Nested model comparison

We can also assess whether a particular subset of *q* parameters is 0

H_0: $\beta_h = \beta_i = \ldots = \beta_g = 0$

To do this we:

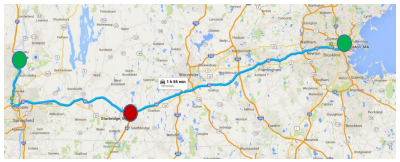
1. Fit the model without these features
2. Calculate the SSE_0 for the model without these predictors
3. Compare it to the full model SSE with an F-statistic:

$$F = \frac{(SSE_0 - SSE)/q}{SSE/(n-k-1)}$$

where q is the number of additional terms in the full model

Enough baseball, back to cars…

Linear regression continued…



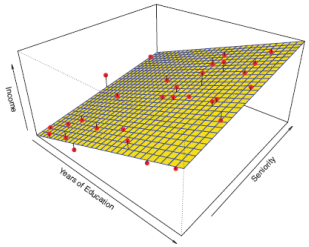
edmunds.com SM

Review of homework 8: multiple linear regression

Multiple linear regression is a linear model that uses more than one feature

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$

price_bought ≈ β_0 + β_1 · mileage_bought
+ β_2 · years_old
+ β_3 · msrp_bought



Used cars from Edmunds data

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	11,155	203	55	~0
mileage_bought	-0.047	0.00356	-13	~0
years_old	-447.22	40	-11	~0
msrp_bought	0.608	0.0049	123	~0

Coefficient of determination R^2 for multiple regression

The **coefficient of multiple determination (R^2)** is a statistic that shows the proportion of the total variance explained by the model

$$R^2 = \frac{SSModel}{SSTotal} = 1 - \frac{SSE}{SSTotal} = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \hat{y})^2}{\Sigma_{i=1}^{n}(y_i - \bar{y})^2}$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \ldots + \hat{\beta}_k \cdot x_k$$

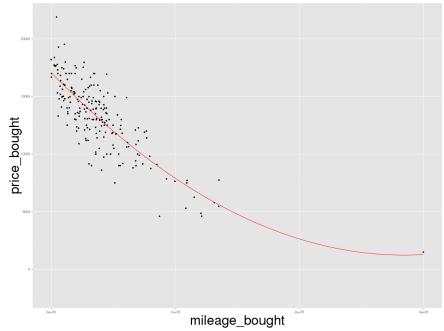
Non-linear relationships

Polynomial regression extends linear regression to non-linear relationships by including nonlinear transformations of covariates

$$\text{price\_bought} = \beta_0 + \beta_1 \cdot \text{mileage\_bought} + \beta_2 \cdot (\text{mileage\_bought})^2 + \varepsilon$$

(still a linear equation but non-linear in original predictors)

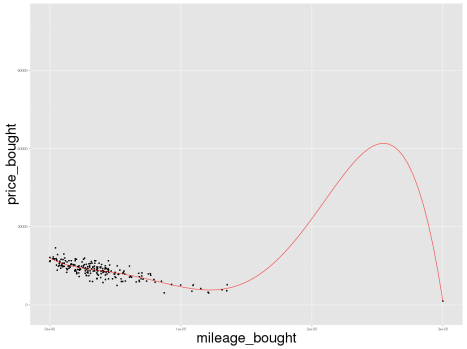
Non-linear relationships on used Toyota Corollas



	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.71E+04	2.40E+02	71.266	< 2e-16 ***
mileage_bought	-1.11E-01	7.22E-03	-15.406	< 2e-16 ***
(mileage_bought)²	1.95E-07	3.47E-08	5.629	4.93e-08 ***

Non-linear relationships on used Toyota Corollas

Do you think the fit will improve if we use higher order polynomials?



Degree	1	2	3	4	5
R^2	0.596197	0.642271	0.6425	0.643618	0.654087

Let's try it in R…

Assessing model fit

To assess model fit we can compute:

R^2 always increases with more predictors x_i because $\hat{y}$ can always fit better with more x_i

1. The coefficient of multiple determination (R^2)

$$R^2 = \frac{SSModel}{SSTotal} = 1 - \frac{SSE}{SSTotal} = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \hat{y})^2}{\Sigma_{i=1}^{n}(y_i - \bar{y})^2}$$

2. The standard deviation of errors $\hat{\sigma}_\varepsilon$

$$\hat{\sigma}_\epsilon = \sqrt{\frac{1}{n-k-1}SSE} = \sqrt{\frac{1}{n-k-1}\Sigma_{i=1}^{n}(y_i - \hat{y})^2}$$

$\hat{\sigma}_\varepsilon$ does **not** always increase with more predictors x_i because of the k in the denominator

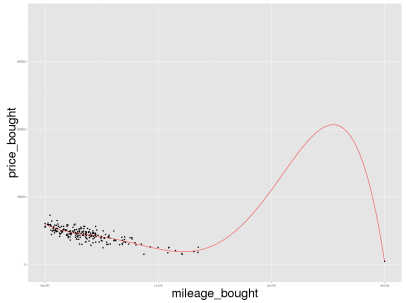
Adjusted R^2

The **adjusted R^2** helps account for the number of predictors in the model

$$R^2_{adj} = 1 - \frac{SSE/(n-k-1)}{SSTotal/n-1} = 1 - \frac{\hat{\sigma}^2_\epsilon}{s^2_y}$$

Thus the adjusted R^2 will not always say that a model with more predictors is a "better" fit to the data

Non-linear relationships on used Toyota Corollas



Second highest adjusted R^2

Highest adjusted R^2

Degree	1	2	3	4	5
R^2	0.596	0.642	0.642	0.644	0.654
Adjusted R^2	0.594	0.639	0.638	0.637	0.647

Are any of the predictors x_i related to y?

```
(Intercept)         X1B         X2B         X3B          HR          BB          AB
106.4370957   0.6380842   0.8382352   1.3246222   1.5579544   0.3496647  -0.1267687

Call:
lm(formula = R ~ X1B + X2B + X3B + HR + BB + AB, data = team_batting)

Residuals:
    Min      1Q  Median      3Q     Max
-76.685 -15.609  -0.977  14.913  79.020

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 106.43710   73.32426   1.452    0.147
X1B           0.63808    0.01678  38.020  < 2e-16 ***
X2B           0.83824    0.02437  34.393  < 2e-16 ***
X3B           1.32462    0.07352  18.018  < 2e-16 ***
HR            1.55795    0.02812  55.406  < 2e-16 ***
BB            0.34966    0.01073  32.598  < 2e-16 ***
AB           -0.12677    0.01618  -7.835 1.09e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.45 on 1116 degrees of freedom
Multiple R-squared:  0.9294,    Adjusted R-squared:  0.929
F-statistic:  2449 on 6 and 1116 DF,  p-value: < 2.2e-16
```

summary(lm_fit)

Better measures of a ‘good’ model

We will discuss other measures for comparing models soon…

Multiple linear regression continued

Overview

More data wrangling with dplyr

Multiple linear regression continued

Final projects!

Create a 5-8 page R Markdown report where you analyze your own data

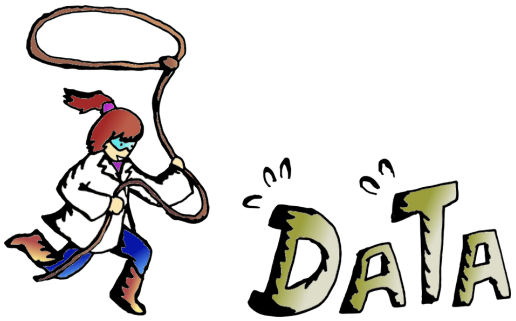
- A chance to practice everything you've learned in class!

Project is due at the end of reading period: ~~December 11th or~~ Dec 8th better?

Sources for data are listed on Canvas

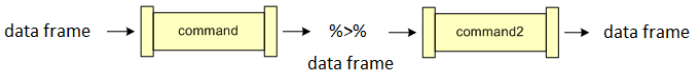
A final project template describing sections in the project is on the class GitHub repository in the homework directory

More data wrangling with dplyr



The pipe operator

The pipe operator %>% allows us to chain commands together




```
car_transactions %>%
        filter(new_or_used_bought == "U")    %>%
        nrow()
```

The summarize() function

The summarize() function reduces values in many rows into single values

```
> car_transactions %>%
      summarize(mean_price = mean(price_bought, na.rm = TRUE))
```

The group_by() function

The group_by() function groups variables for future operations

```
> car_transactions %>%
      group_by(make_bought)

# It is only useful in conjunction with other functions
> car_transactions %>%
      group_by(make_bought)  %>%
      summarize(mean_price = mean(price_bought, na.rm = TRUE))
```

Bentley's are expensive!

Let's try it in R…

Homework 9: Flight delays ☹

Data set contains information about flights leaving NYC in 2013

```
> library("nycflights13")
> data(flights)
```

Back to multiple regression…

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \ldots + \hat{\beta}_k \cdot x_k$$

Qualitative predictors

Predictors can be categorical as well as quantitative

If a predictor only has two levels, we can use a single 'dummy variable' to encode these two levels:

- E.g., used or new car

$$x_i = \begin{cases} 1 & \text{if the } i\text{th car is new} \\ 0 & \text{if the } i\text{th car is used} \end{cases}$$

New cars have an additional value added to their y-intercepts

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if the } i\text{th car is new} \\ \beta_0 + \varepsilon_i & \text{if the } i\text{th car is used} \end{cases}$$

Qualitative predictors

When a qualitative predictor has k levels, we need to use k -1 dummy variables to code it

- E.g., suppose cars could be red, black or white, then we can code for color using two dummy variables

$$x_{i1} = \begin{cases} 1 & \text{if the } i\text{th car is red} \\ 0 & \text{if the } i\text{th car is not red} \end{cases}$$

$$x_{i2} = \begin{cases} 1 & \text{if the } i\text{th car is black} \\ 0 & \text{if the } i\text{th car is not black} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if the ith car is red} \\ \beta_0 + \beta_2 + \varepsilon_i & \text{if the ith car is black} \\ \beta_0 + \varepsilon_i & \text{if the ith car is white} \end{cases}$$

Interaction terms

An ***interaction effect*** occurs when the response variable y is influenced by the levels of two or more predictors in an non-additive way

For example, the price of a car might be more effected by the miles driven depending on the model of the car

- E.g., the price of a BMW could depreciate faster with more mileage than for a Mazda

We can model this using an equation with an interaction term

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 (x_1 \cdot x_2) + \epsilon$$

Interaction terms

When using categorical variables, the interaction corresponds to ***different slopes*** for the quantitative variable depending on the value of the categorical variable

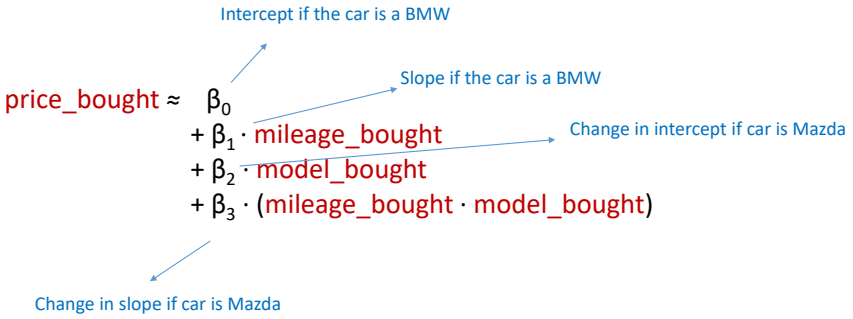
If new: $\text{price\_bought} \approx \beta_0 + \beta_1 \cdot \text{mileage\_bought}$

If used: $\text{price\_bought} \approx (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \cdot \text{mileage\_bought}$

Additive term if car is Mazda

Change in slope if car is Mazda

Interaction terms – full model



Remember, you should check conditions (i.e., create diagnostic plots) when doing inference on regression models

Let's try it in R…

Model selection methods and the grammar of graphics

Overview

Model selection methods

Brief history of data visualization

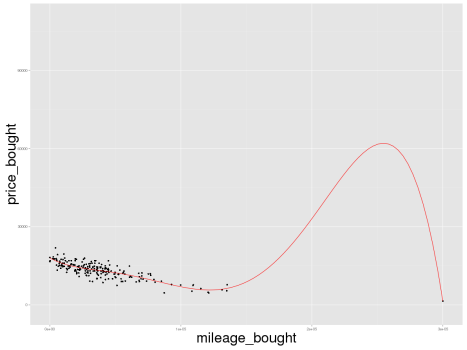
The grammar of graphics and ggplot

Any questions about homework 9?

Simple and complex models

Review: Non-linear relationships on used Toyota Corollas

Is the 5th degree polynomial fit a "good model"?

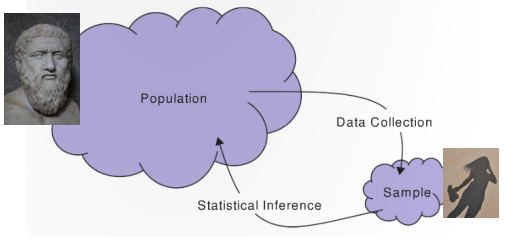


Degree	1	2	3	4	5
R^2	0.596197	0.642271	0.6425	0.643618	0.654087

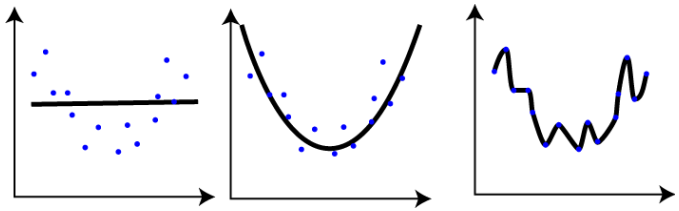
Overfitting

Overfitting occurs when we generate a function that too closely matches random sample we have, but does not generalize to the full probability distribution

- The model is fit to closely to the shadows and not getting at the Truth



Overfitting



Overfitting example

In homework 8, problem 2.4 you tried to get the highest value for R^2 for predicting car price

Ryan Schiller's model:

$$\hat{y} = 24919.66 + 101976420913.07MS - 8801176308.15MS^2 + 1241642151.11MS^3 + -294845652.56MS^4 + 81081110.14MS^5 - 19486206.21MS^6 + 4627457.40MS^7 - 962171.87MS^8 + 159411.19MS^9 + 4758.12MS^10 - 215919.33YO + 61629.11YO^2 - 46883.91YO^3 + 27616.00YO^4 + -41.24YO^5 + 986604.79MB - 35074977.23MB^2 - 65070922.16MB^3 - 80975553.98MB^4 - 39704725.17MB^5 - 13029640.26MB^6 - 4181962.33MB^7 - 1234301.18MB^8 - 444859.17MB^9 - 130775.72MB^10 + 10404.48MPY - 3884.70MPY^2 + 2328.80MPY^3 + 13563.89MPY^4 - 5277.89MPY^5 - 15338.18MPY^6 + 12851.91MPY^7 + 4899.51MPY^8 + 383.03MPY^9 - 68.52MPY^10 - 82637866000.10(log(MS)) - 41355241329.94(log(MS))^2 - 32425545929.48(log(MS))^3 - 2481236288.29(log(MS))^4 - 1497429367 4.01(log(MS))^5 - 762403555 7.33(log(MS))^6 - 226487189 0.73log(MS)^7 - 411050984.04log(MS)^8 - 54557306.69log(MS)^9 - 2606070.61log(MS)^10 + 172377.75(YO * MS) - 85365.06(YO * MS)^2 + 20333.05(YO * MS)^3 + 38388.69(YO * MS)^4 - 11361.46(YO * MS)^5 + 37428.48(YO * MS)^6 + 5120.32(YO * MS)^7 + 15285.89(YO * MS)^8 + 9300.88(YO * MS)^9 + 11552.40(YO * MS)^10 + 35723937.44w^2 + 66215674.21w^3 + 80347987.00w^4 + 38445702.35w^5 + 12868623.19w^6 + 4105787.54w^7 + 1161535.74w^8 + 426310.43w^9 + 133553.84w^10$$

R^2 = 0.9434

https://www.youtube.com/watch?v=DQWI1kvmwRg

Ways to deal with overfitting

1. Creating measures of fit (statistics) that penalize models with more predictors

2. Creating models that try to shrink the magnitude of the coefficients

3. Creating simpler models by removing predictors

4. Evaluating models using cross-validation

Review: statistics that penalize larger models

To assess model fit we can compute:

R^2 always increases with more predictors x_i because $\hat{y}$ can always fit better with more x_i

1. The coefficient of multiple determination (R^2)

$$R^2 = \frac{SSModel}{SSTotal} = 1 - \frac{SSE}{SSTotal} = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \hat{y})^2}{\Sigma_{i=1}^{n}(y_i - \bar{y})^2}$$

2. The standard deviation of errors $\hat{\sigma}_\epsilon$

$$\hat{\sigma}_\epsilon = \sqrt{\frac{1}{n-k-1}SSE} = \sqrt{\frac{1}{n-k-1}\Sigma_{i=1}^{n}(y_i - \hat{y})^2}$$

$\hat{\sigma}_\epsilon$ does **not** always increase with more predictors x_i because of the k in the denominator

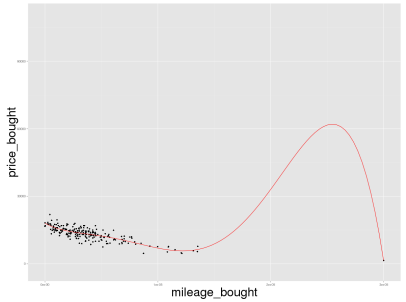
Review: Adjusted R^2

The **adjusted R^2** helps account for the number of predictors in the model

$$R^2_{adj} = 1 - \frac{SSE/(n-k-1)}{SSTotal/n-1} = 1 - \frac{\hat{\sigma}^2_\epsilon}{s^2_y}$$

Thus the adjusted R^2 will not always say that a model with more predictors is a "better" fit to the data

Review: Non-linear relationships on used Toyota Corollas



According to the adjusted R^2 statistic, the degree 5 model is best, which doesn't seem right

Degree	1	2	3	4	5
R^2	0.596	0.642	0.642	0.644	0.654
Adjusted R^2	0.594	0.639	0.638	0.637	0.647

Statistics that penalize larger models

There are several other statistics that also penalize models that have more features

- These statistics are only meaningful for within data set comparisons

Akaike information criterion: $AIC = 2 \cdot k + n \cdot ln(SSE)$ R: AIC(lm_fit)

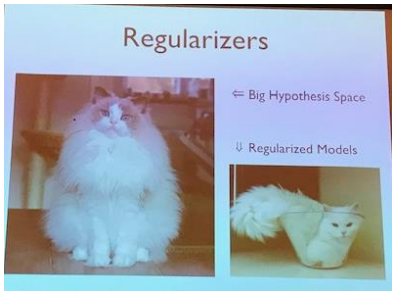
Bayesian information criterion: $BIC = k \cdot ln(n) + n \cdot ln(SSE/n)$ R: BIC(lm_fit)

These models penalize models with more predictors

One should select the model with the lowest value on these statistics

Brief mention: shrinkage methods

Rather than finding the coefficients $\hat{\beta}_i$ by just minimizing the SSE, one can also add penalties in the fitting procedure to find simpler models



We will very briefly discuss two techniques:

- Ridge regression (l_2 norm penalty)
- The lasso (l_1 norm penalty)

Brief mention: Ridge regression

Ridge regression finds the coefficients β_i's that minimize:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

SSE — shrinkage penalty — Tuning parameter

What happens if:

- $\lambda = 0$
- $\lambda \rightarrow$ infinity

(the coefficients depend on the tuning parameter value)

Brief mention: The Lasso

The Lasso finds the coefficients β_i's that minimize:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

SSE — shrinkage penalty — Tuning parameter

Similar to ridge regression but penalizes $|\beta_j|$ instead of β_j^2

- i.e., uses the l_1 penalty instead of the l_2 penalty

Advantages

- Final model will often have many set β_j to 0
- i.e., does variable selection and creates a 'sparse' model

Shrinkage methods

Shrinkage methods can lead to much better predictions on new data

- They induce a bias in the model but it greatly cuts down on the model variance
  - e.g., deals with some of instability seen with multicollinearity

Take a machine learning class to learn more!

Brief mention: Variable selection

Variable selection refers to finding models that rely on a small subset of predictors

- This can help make the regression model more interpretable as well

We could use individual feature p-values to determine which predictors to use, however…

- The p-values change as predictors are added and removed
  - Due to multicollinearity

Feature selection: deciding which variables to use

Ideally we would like to try all combinations of predictors, however, if there are k features, there are 2^k possible models which can be intractable

A few heuristic methods exist for selecting smaller models

- Forward selection: start with a model with no predictors and add predictors (until you have enough)
- Backward selection: Start with the full model and delete predictors
- Mixed selection: Use a combination of forward and backward selection

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
X1B   1.355e+00  5.700e-01   2.378   0.0176 *
X2B   3.170e-01  1.786e+00   0.177   0.8592
X3B   4.598e+00  6.128e+00   0.750   0.4532
HR    4.143e+00  2.041e+00   2.030   0.0426 *
BB   -2.925e-01  8.564e-01  -0.342   0.7328
X1Bn -6.359e+03  3.150e+03  -2.019   0.0437 *
X2Bn  9.474e+02  9.871e+03   0.096   0.9236
X3Bn -2.001e+04  3.393e+04  -0.590   0.5555
XHRn -1.683e+04  1.128e+04  -1.492   0.1359
XBBn  3.600e+03  4.732e+03   0.761   0.4469
```

In R: leaps::regsubsets()

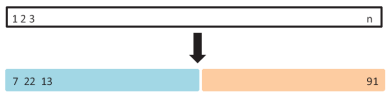
Cross-validation

To realistically assess how accurate your predictions are on new data we can use cross-validation

Cross-validation consists of splitting your data into two sets

A training set in which the parameters of classification/regression model are fit

A test set in which the prediction accuracy of our model is assessed



Mean squared prediction error

To evaluate how effective a model is, we can use the mean squared prediction error (MSPE) using the following steps:

1. Fit a model using the training data
2. Make predictions on the test data
3. Calculate the MSPE on the test data:

$$MSPE = \frac{1}{n_t}\sum_{i=1}^{n_t}(y_i - \hat{y}_i)^2$$

Actual y values in the test set

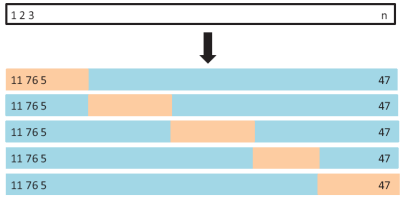
Predicted y values in the test set

n_t is the number of points in the test set

K-fold cross-validation

K-fold cross-validation

- Split the data into k parts
- Train on k-1 of these parts and test on the left out part
- Repeat this process for all k parts
- Average the prediction accuracies to get a final estimate of the generalization error



Leave-one-out (LOO) cross-validation: k = n

Let's try it in R...

Data visualization!

What are some reasons we visualize data rather than just reporting statistics?

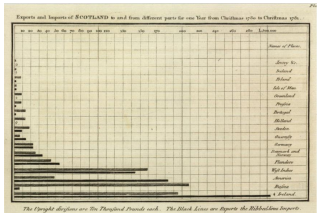
Whatever relates to extent and quantity may be represented by geometrical figures. Statistical projections which speak to the senses without fatiguing the mind, possess the advantage of fixing the attention on a great number of important facts

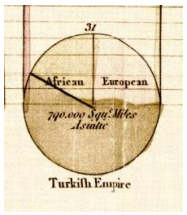
—Alexander von Humboldt, 1811

A very brief history of data visualization

The age of modern statistical graphs began around the beginning of the 19th century

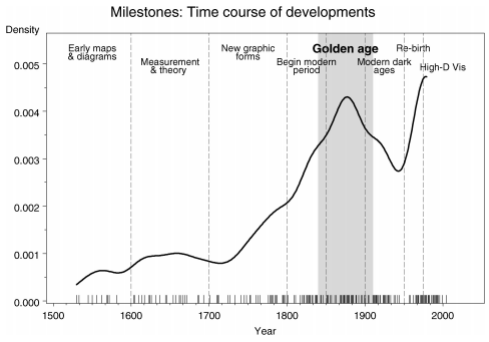
William Playfair (1759-1823) credited with inventing the line graph, bar chart and pie chart





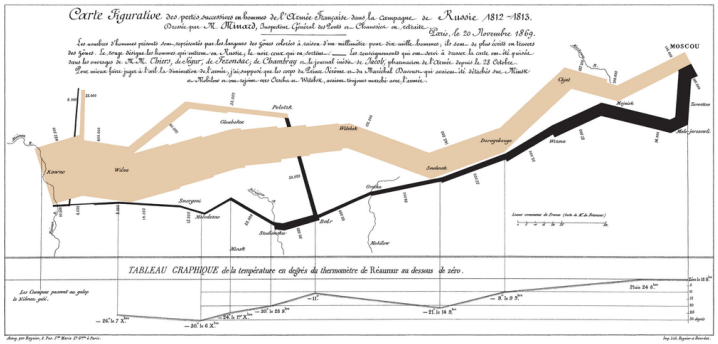
A very brief history of data visualization

According to Friendly, statistical graphics researched its golden age between 1850-1900



A very brief history of data visualization

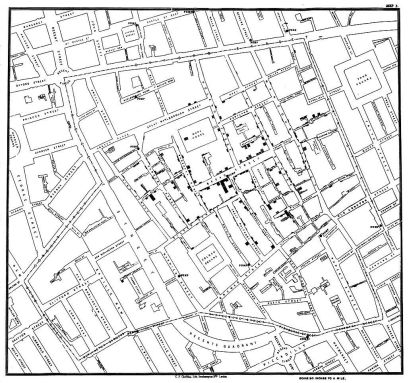
Joseph Minard (1781-1870)



Map of Napoleon's march on Russia

A very brief history of data visualization

John Snow (1813-1858)



Clusters of cholera cases in London epidemic of 1854

A very brief history of data visualization

Florence Nightingale (1820-1910)

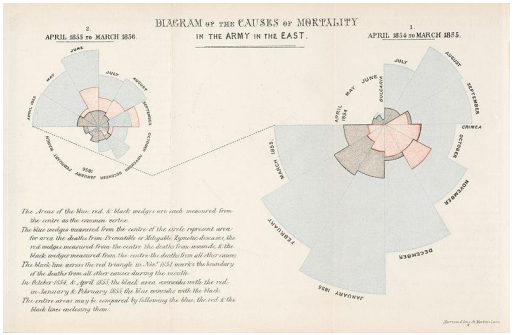
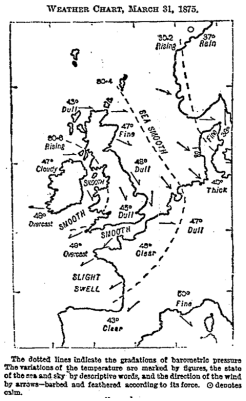


Diagram of the causes of mortality in the army in the east

A very brief history of data visualization

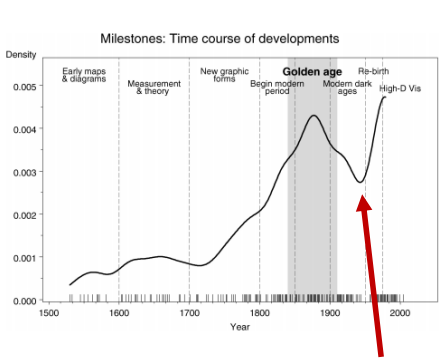
Francis Galton (1822-1911)

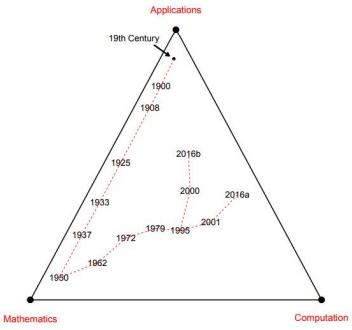


First weather map published in a newspaper (1875)

A very brief history of data visualization

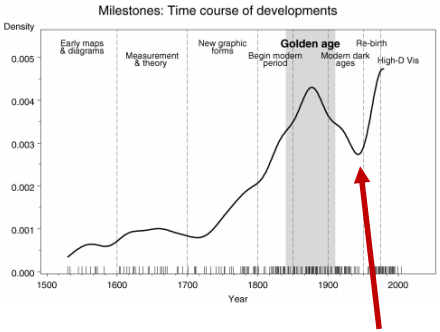
"Graphical dark ages" around 1950

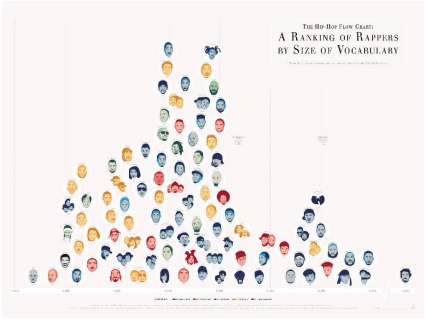




A very brief history of data visualization

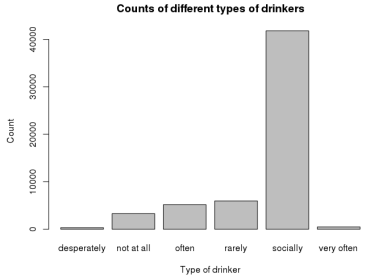
Currently undergoing a “Graphical re-birth”



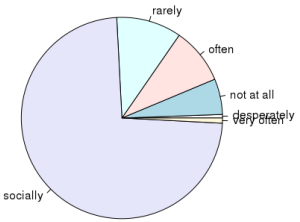


Review: plots of categorical data

Bar plot

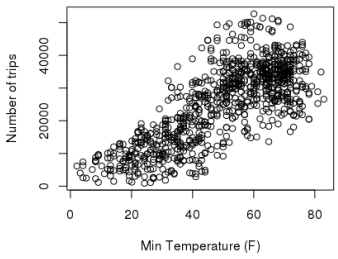


Pie chart

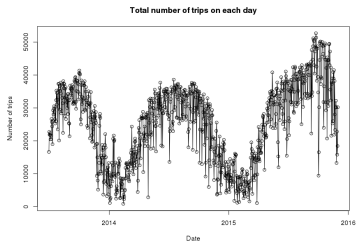


Review: plots of quantitative data

Scatter plots

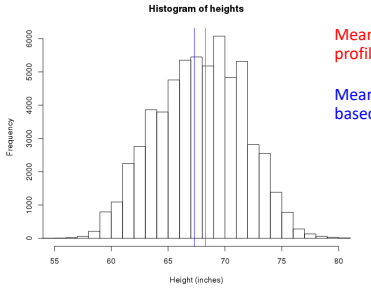


Line chart



Review: plots of quantitative data

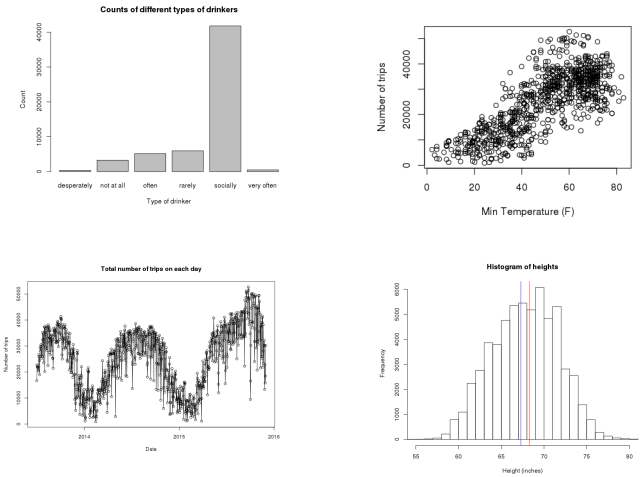
Histograms



Mean height in the profiles data

Mean height expected based on US population

What are some similarities between these graphs?

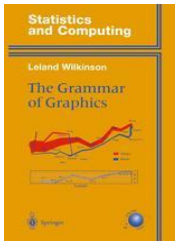


The grammar of graphics

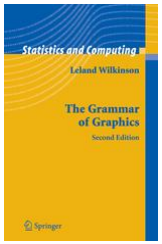
Leland Wilkinson noticed similarities between many graphs and tried to generate a 'grammar' that could be used to express a graph

- i.e., a list elements that can be combined together to create a graph

First edition



Second edition



Graphs are composed of…

A Frame: Coordinate system on which data is placed

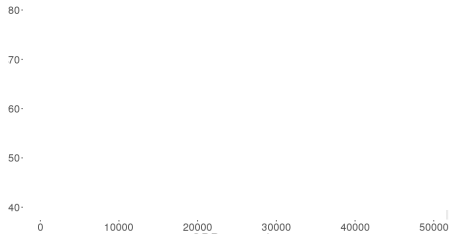
- E.g., Cartesian coordinate system, polar coordinates, etc.

Glyphs: basic graphic unit representing cases or statistics

- Contains visual properties (aesthetics) such as: shape, color, size, etc.
- Need to specify how properties of the data are mapped onto these aesthetics

Scales and guides: shows how to interpret axes and other properties of the glyphs

- i.e., gives information about how the data values were mapped into glyph properties



Plots can also contain…

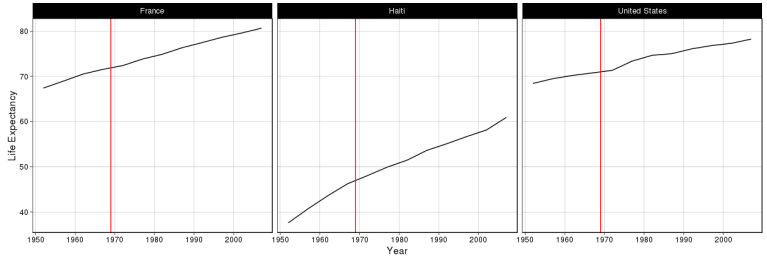
Facets: allows for multiple side-by-side graphs based on a categorical variable

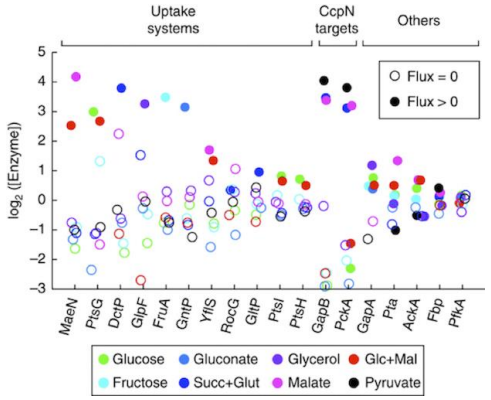
- Makes it easier to compare different conditions

Layers: allows for more than one types of data to be mapped onto the same figure

Theme: contains finer points of display

- E.g., font size, background color, etc.





The variables are:

- Log enzyme concentration
  - -3 to 5
- Target
  - CcpN, Uptake,…
- Flux
  - Zero or positive
- Gene
  - MaeN, PtsG, …
- Molecule:
  - Glocose, Fructose, …

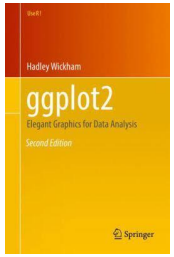
1. What are the guides, and the mapping between variable and visual attributes?
2. What are the graphical attributes of the glyph?
3. Which variables set the frame?
4. Can you reconstruct the data frame that underlies this figure?

ggplot

ggplot2 is an R package that implements the grammar of graphics

- It builds up graphics by starting with a frame, adding glyphs, etc.

```
# load the ggplot2 library
> library('ggplot2')
```



Get the book on GitHub

Model selection methods and the grammar of graphics

Overview

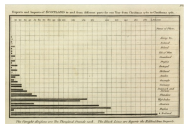
The grammar of graphics and ggplot

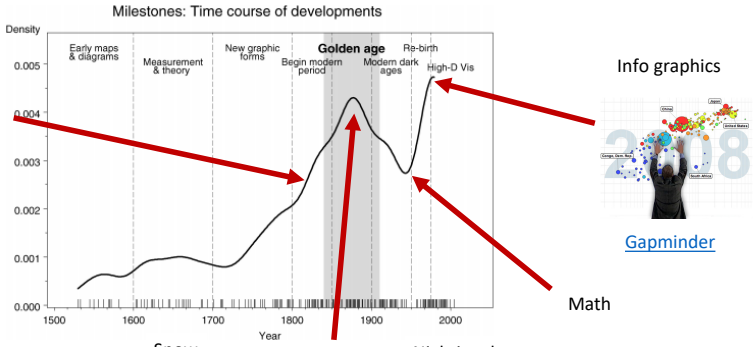
Joining data tables

If time: reshaping data

Review: A very brief history of data visualization

Playfair



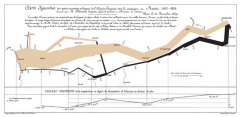


Info graphics

Gapminder

Math

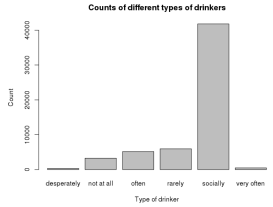
Minard

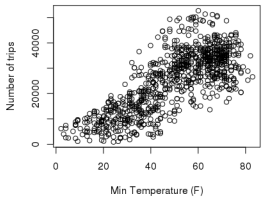


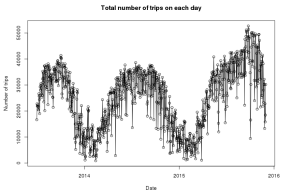
Snow

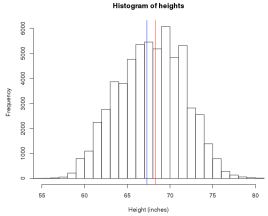
Nightingale

What are some similarities between these graphs?







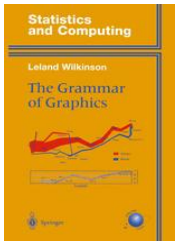


The grammar of graphics

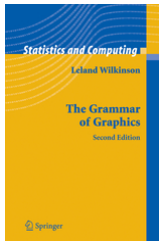
Leland Wilkinson noticed similarities between many graphs and tried to generate a 'grammar' that could be used to express a graph

- i.e., a list elements that can be combined together to create a graph

First edition



Second edition



Graphs are composed of…

A Frame: Coordinate system on which data is placed

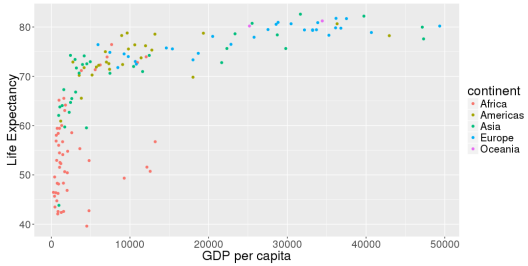
- E.g., Cartesian coordinate system, polar coordinates, etc.

Glyphs: basic graphic unit representing cases or statistics

- Contains visual properties (aesthetics) such as: shape, color, size, etc.
- Need to specify how properties of the data are mapped onto these aesthetics

Scales and guides: shows how to interpret axes and other properties of the glyphs

- i.e., gives information about how the data values were mapped into glyph properties

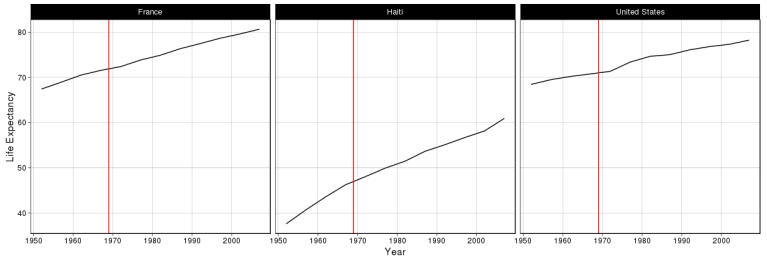


Plots can also contain…

Facets: allows for multiple side-by-side graphs based on a categorical variable
- Makes it easier to compare different conditions

Layers: allows for more than one types of data to be mapped onto the same figure

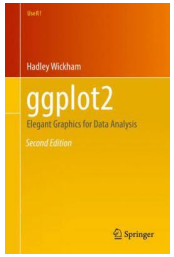
Theme: contains finer points of display
- E.g., font size, background color, etc.



ggplot

ggplot2 is an R package that implements the grammar of graphics
- It builds up graphics by starting with a frame, adding glyphs, etc.

```
# load the ggplot2 library
> library('ggplot2')
```



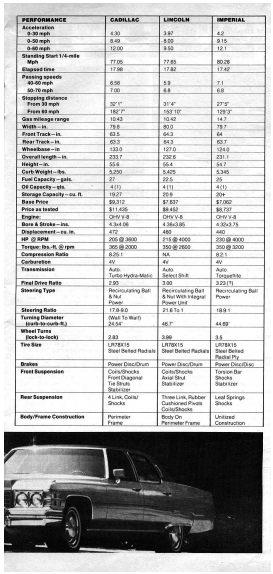
Get the book on GitHub

Example data: mtcars


ROAD TEST By Jim Brokaw

THE LUXURY CARS

Imperial Palace
Fortress Fleetwood
Castle Continental



PERFORMANCE	CADILLAC	LINCOLN	IMPERIAL
Acceleration			
0-30 mph	4.30	3.97	4.2
0-50 mph	8.49	8.00	9.15
0-60 mph	12.00	9.50	12.1
Standing Start 1/4-mile			
Mph	77.05	77.85	80.28
Elapsed time	17.98	17.82	17.42
Passing speeds			
40-60 mph	6.56	5.9	7.1
50-70 mph	7.00	6.8	6.9
Stopping distance			
From 30 mph	32'1"	31'4"	27'3"
From 60 mph	192'7"	152'10"	129'2"
Gas mileage range	10.43	10.42	14.7
Width—in.	78.6	80.0	79.7
Front Track—in.	63.5	64.3	64
Rear Track—in.	63.3	64.3	63.7
Wheelbase—in	133.0	127.0	124.0
Overall length—in.	233.7	232.6	231.1
Height—in.	54.6	54.4	54.7
Curb Weight—lbs.	5,230	5,425	5,345
Fuel Capacity—gals.	27	22.5	25
Oil Capacity—qts.	4 (1)	4 (1)	4 (1)
Storage Capacity—cu. ft.	19.27	20.9	20+
Base Price	$9,312	$7,837	$7,062
Price as tested	$11,435	$9,432	$9,737
Engine:	OHV V-8	OHV V-8	OHV V-8
Bore & Stroke—ins.	4.3x4.06	4.36x3.85	4.30x3.75
Displacement—cu. in.	472	480	440
HP @ RPM	205 @ 3800	215 @ 4000	220 @ 4000
Torque: lbs.-ft. @ rpm	365 @ 2000	350 @ 2600	350 @ 3200
Compression Ratio	8.05:1	NA	8.0:1
Carburetion	4V	4V	4V
Transmission	Auto. Turbo Hydra-Matic	Auto. Select Shift	Auto. TorqueFlite
Final Drive Ratio	2.93	3.00	3.23 (?)
Steering Type	Recirculating Ball & Nut Power	Recirculating Ball & Nut With Integral Power Unit	Recirculating Ball Power
Steering Ratio	17.9-9.0	21.6 To 1	18.9:1
Turning Diameter (curb-to-curb-ft.)	(Wall To Wall) 24.34'	46.7'	44.69'
Wheel Turns (lock-to-lock)	2.83	3.66	3.5
Tire Size	LR78X15 Steel Belted Radials	LR78X15 Steel Belted Radials	LR78X15 Steel Belted Radial TV
Brakes	Power Disc/Drum	Power Disc/Drum	Power Disc/Disc
Front Suspension	Coils/Shocks Front Diagonal Tie Struts Stabilizer	Coils/Shocks Axial Strut Stabilizer	Torsion Bar Shocks Stabilizer
Rear Suspension	4 Link, Coils/ Shocks	Three Link, Rubber Cushioned Pivots Coils/Shocks	Leaf Springs Shocks
Body/Frame Construction	Perimeter Frame	Body On Perimeter Frame	Unitized Construction

mtcars data frame

What variables are in a data frame?

```
> View(mtcars)      # only works in Rstudio, not in Markdown
> glimpse(mtcars)
> ? mtcars      # this data frame as a code book
```

[, 1]	mpg	Miles/(US) gallon
[, 2]	cyl	Number of cylinders
[, 4]	hp	Gross horsepower
[, 6]	wt	Weight (1000 lbs)
[, 9]	am	Transmission (0 = automatic, 1 = manual)

Let's try it in R…

Do cars that weigh more use more fuel?

Question: do cars that weigh more use more fuel?

What variables in the mtcars data frame are of interest?

- mpg
- wt

We can create a scatter plot using base graphics…

```
> plot(mtcars$wt, mtcars$mpg)
```

Creating a scatter plot in ggplot

Frame (coordinate system)

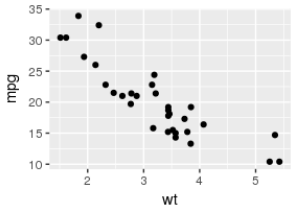
Adds a layer with glyphs

Aesthetic mapping

```
> ggplot(data = mtcars) +
      geom_point(mapping = aes(x = wt, y = mpg))

# alternatively (sets aesthetic mapping globally)
> ggplot(data = mtcars, mapping = aes(x = wt, y = mpg)) +
      geom_point()

# a shorter version:
> ggplot(mtcars, aes(x = wt, y = mpg)) +
      geom_point()
```

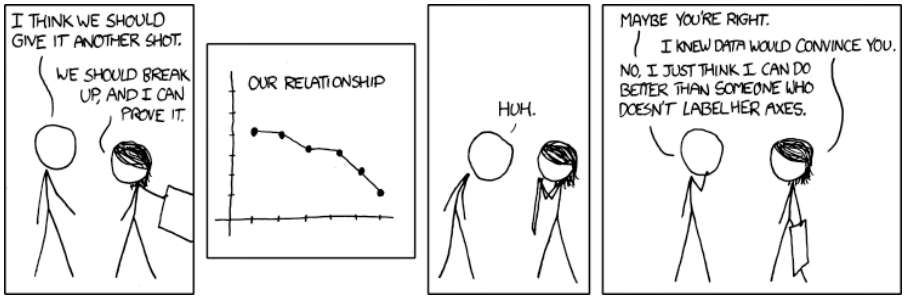


Adding labels to plots

We can add labels to the plots using the xlab("label1") and ylab("label2") functions

Add labels to your last plot

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
      geom_point()  +
      xlab("Weight") +
      ylab("Miles per Gallon")
```



If you don't want an ex, label you axes!

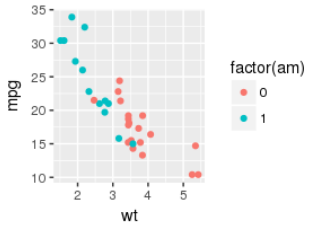
More aesthetic mappings

Let's look at the relationship between weight, miles per gallon and transmission type on the same graph by plotting… (?)

```
> ggplot(mtcars, aes(x = wt, y = mpg, col = am)) +
        geom_point()

# It is better if we make am a categorical variable
> ggplot(mtcars, aes(x = wt, y = mpg, col = factor(am))) +  geom_point()
```

Notice the guides!!!



Try mapping am on to shape using:

1. *shape = am*
2. size using: *size = am*

Which is better to use color or shape or size?

Attributes vs. Aesthetics

Setting **aesthetics** map a variable to a glyph property

Setting **attributes** set a glyph property to a fixed value

```
# setting a aesthetic
> ggplot(mtcars) +
        geom_point(aes(x = wt, y = mpg, col = factor(am)))

# setting an attribute
> ggplot(mtcars) +
        geom_point(aes(x = wt, y = mpg), col = "red")
```

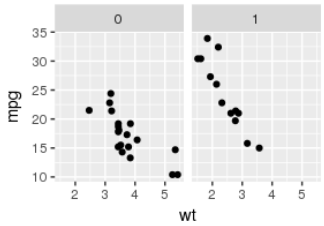
Outside the aesthetic mapping!

Facets

Beyond comparing variables based on aesthetics you can compare categorical variables by splitting a plot into subplots (called facets) using facet_wrap

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +  geom_point() +
        facet_wrap(~am)
```

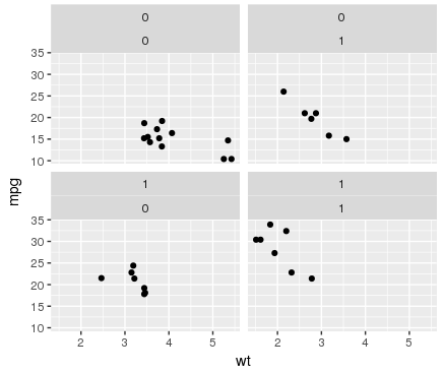
What do facets make it easy to see on this graph?



Facets along two dimensions

One can also do facets in two dimensions

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point() +
        facet_wrap(vs ~am)
```



Exploring graph properties

Gapminder tools: https://www.gapminder.org/tools

```
> library('gapminder')
```

Overplotting

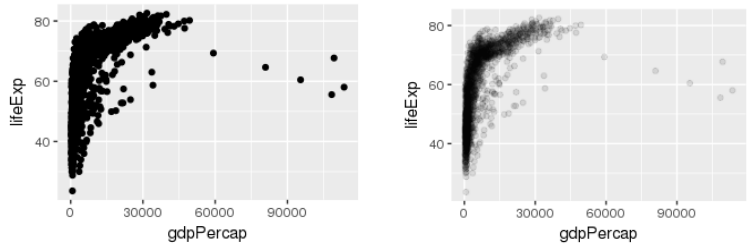
Sometimes points overlap making it hard to estimate the number of points at a particular range of values

We can control the transparency of points by changing their alpha values

```
# compare these two plots
> ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
        geom_point()

> ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
        geom_point(alpha = .1)
```

Overplotting



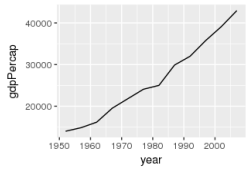
Geometries: line plot

So far we've only created scatter plots, but we can use different geoms to create other types of plots

Create a plot that shows the GDP in the United States as a function of the year using the geom *geom_line()*

- Hint: filter the gapminder data first…

```
> gapminder %>% filter(country == 'United States') %>%
      ggplot(aes(x = year, y = gdpPercap)) +
      geom_line()
```



Geometries: line plot

We can also make histograms using the geom_histogram() function.

Plot a histogram of the weights of cars

```
> ggplot(mtcars, aes(x = wt)) +
      geom_histogram()
```

Note the histogram geom only has an x aesthetic, and does not have a y aesthetic value.

Geometries: boxplot

There are many other geom as well, including geom_boxplot()

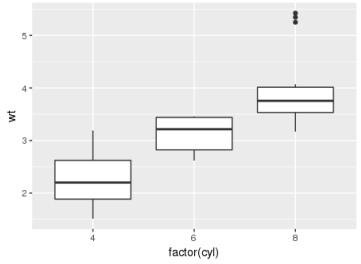
Plot a boxplot of the weights of cars

```
> ggplot(mtcars, aes(x = "", y = wt)) +
      geom_boxplot()
```

Side-by-side boxplots

Often it is useful to compare boxplots across different groups

```
> ggplot(mtcars, aes(x = factor(cyl), y = wt)) +
      geom_boxplot()
```



Violin and Joy plots

Violin and Joy plots are other ways to view distributions of data

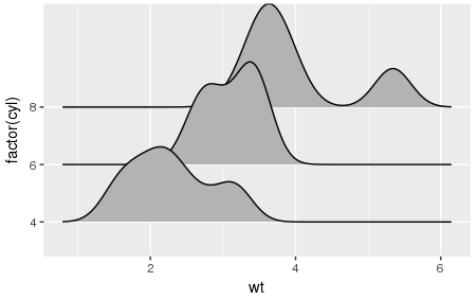
```
> ggplot(mtcars, aes(x = factor(cyl), y = wt)) +
      geom_violin()
```

Note x and y are reversed

```
> library("ggridges")
> ggplot(mtcars, aes(y = factor(cyl), x = wt)) +
      geom_density_ridges()
```

Violin and Joy plots

Any ideas why they are called joy plots?



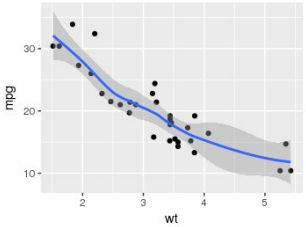
Multiple layers

We can also have multiple geom layers on a single graph by using the + symbol

- E.g ggplot(…) + geom_type1() + geom_type2()

Create a scatter plot of miles per gallon as a function of weight and then add a smoothed line using geom_smooth()

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
      geom_point() +
      geom_smooth()
```



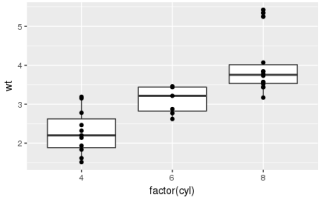
Multiple layers

We can also have multiple geom layers on a single graph by using the + symbol

- E.g ggplot(…) + geom_type1() + geom_type2()

Recreate a boxplot of weight (wt) grouped by the factor of cylinders (cyl), and then add points using geom_point()

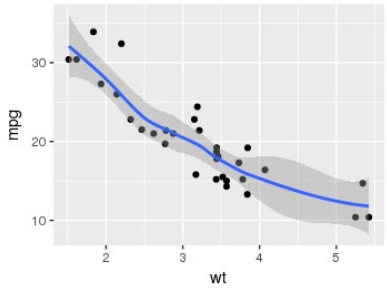
```
> ggplot(mtcars, aes(x = factor(cyl), y = wt)) +
      geom_boxplot() +
      geom_point()
```



Multiple layers

Create a scatter plot of miles per gallon (mpg) as a function of weight (wt) and then add a smoothed line using geom_smooth()

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
      geom_point() +
      geom_smooth()
```



Themes

We can also use different types to change the appearance of our plot

Add theme_classic() to your plot

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
      geom_point()  +
      xlab("Weigth") +
      ylab("Miles per Gallon") +
      theme_classic()
```

Bonus features

Additional geometries: emoGG

There are also additional packages that add more geoms

```
>  library(emoGG)
> ggplot(mtcars, aes(wt, mpg)) +
      geom_emoji(emoji="1f697")
```

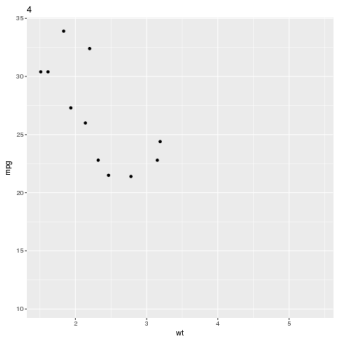
Animation

We can create animated images (gifs) using the gganimate package

```
> library(gganimate)
> library(gapminder)
```

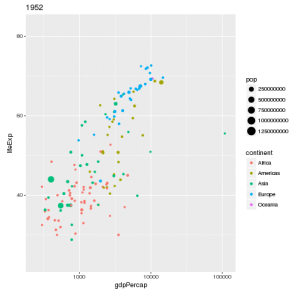
In the gapminder video, Hans had the following mapping:

- x = gpd per capita
- y = life expectancy
- size = population
- color = continent
- frame = year



Recreating gapminder plot

```
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_x_log10() +
  facet_wrap(~continent) +
  # Here comes the gganimate specific bits
  labs(title = 'Year: {frame_time}',
       x = 'GDP per capita', y = 'life expectancy') +
  transition_time(year) +
  ease_aes('linear')
```



Plotly – interactive plots

```
> library(plotly)

p <- ggplot(gapminder, aes(x = gdpPercap, y = lifeExp,
        size = pop, col = continent, frame = year)) +
        geom_point() +
        scale_x_log10()

ggplotly(p)
```

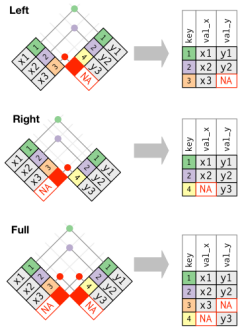
Joining data frames

Often data of intertest is spread across multiple data frames

We can join two data together into a single data frame for further analyses using dplyr

There are several different ways to join tables:

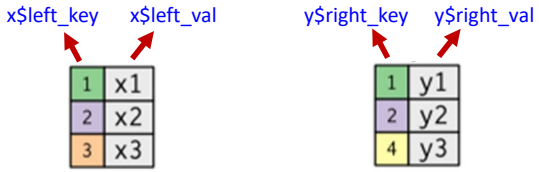
- Left join
- Right join
- Inner join
- Full join
- Etc.



Left and right tables

Suppose we have two data frames called x and y

- x have two variables called left_key, and left_val
- y has two variables called right_key and right_val



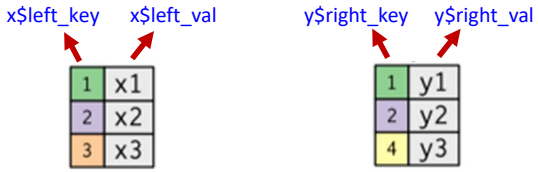
Left table (x) **Right table (y)**

download_class_data('x_and_y.Rda')

Left and right tables

Suppose we have two data frames called x and y

- x have two variables called left_key, and left_val
- y has two variables called right_key and right_val



Left table (x) **Right table (y)**

Joins have the general form:

join(x, y, by = c("left_key" = "right_key"))

Left joins

Left joins keep all rows in the left table.

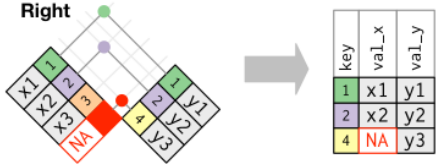
Data from right table added when there is the key matches, otherwise NA as added.

> left_join(x, y, by = c("left_key" = "right_key"))

Right joins

Right joins keep all rows in the right table.

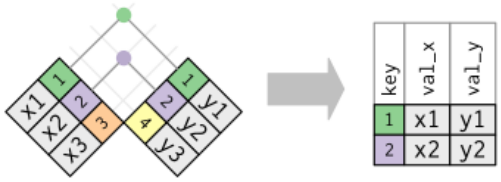
Data from left table added when there is the key matches, otherwise NA as added.



> right_join(x, y, by = c("left_key" = "right_key"))

Inner joins

Inner joins only keep rows in which there are matches between the keys in both tables

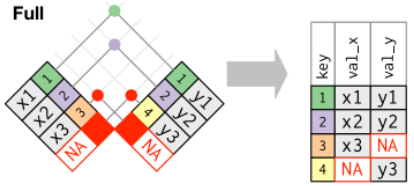



```
> inner_join(x, y, by = c("left_key" = "right_key"))
```

Full joins

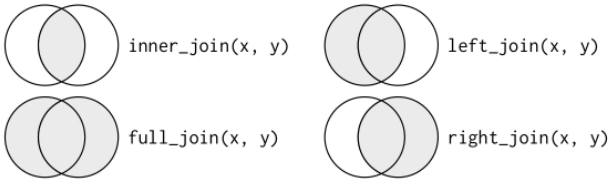
Full joins keep all rows in both table.

NAs are added where there are no matches




```
> full_join(x, y, by = c("left_key" = "right_key"))
```

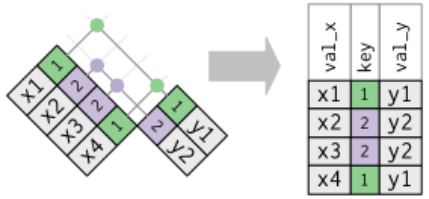
Summary



Duplicate keys

Duplicate keys are useful if there is a one-to-many relationship

- duplicates are usually in the left table

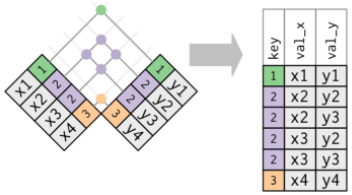


Let's join the whether prediction cityIDs with actual names and other information about these cityID locations

Duplicate keys

If both tables have duplicate keys you get all possible combinations (Cartesian product)

- This is usually an error!



Again, always check the output after you join a table because even if there is not a syntax error you might not get the table you are expecting!

Duplicate keys

To deal with duplicate keys in both tables, we can join the tables using multiple keys in order to make sure that each row is uniquely specified

We can do this using the syntax:

join(x, y, by = c("L_key1" = "R_key1", "L_key2" = "R_key2"))

Duplicate keys

```
> x2 <- data.frame(L_key1 = c(1, 2, 2),
                   L_key2 = c("a", "a", "b"),
                   L_val = c("x1", "x2", "x3"))

> y2 <- data.frame(R_key1 = c(1, 2, 2, 3, 3),
                   R_key2 = c("a", "a", "b", "a", "b"),
                   R_val = c("y1", "y2", "y3", "y2", "y3"))

> inner_join(x2, y2, c("L_key1" = "R_key1"))

> inner_join(x2, y2, c("L_key1" = "R_key1", "L_key2" = "R_key2"))
```

Structured Query Language

As mentioned before, having multiple tables that can be joined together is common in Relational Database Systems (RDBS)

- A common language used by RDBS is Structured Query Language (SQL)

dplyr	SQL
`inner_join(x, y, by = "z")`	`SELECT * FROM x INNER JOIN y USING (z)`
`left_join(x, y, by = "z")`	`SELECT * FROM x LEFT OUTER JOIN y USING (z)`
`right_join(x, y, by = "z")`	`SELECT * FROM x RIGHT OUTER JOIN y USING (z)`
`full_join(x, y, by = "z")`	`SELECT * FROM x FULL OUTER JOIN y USING (z)`

Wrap up

Wrap up

Topics we will cover

~~**R and descriptive statistics/plots**: The basics of base R, fundamental concepts in Statistics~~

~~**Review confidence intervals:** Sampling and bootstrap distributions, t distributions~~

~~**Review of hypothesis tests:** Permutation tests, *non-parametric tests**, theories of testing~~

ANOVA: ~~one-way~~, multi-way, interactions, ~~*mixed effects**~~

~~**Regression:** simple/multiple, non-linear terms~~, logistic regression

~~**Data wrangling:** filtering and summarizing data, joining data sets~~, reshaping data

~~**Data visualization:** grammar of graphics, mapping~~

~~**Statistical learning:** cross-validation, *supervised learning*, PCA*, clustering*,~~

~~***Misc:** text analysis and manipulation*, Bayesian methods*, other topics based on interest~~

**Time permitting*

Overview

More ANOVAs:

- Connections between one-way ANOVA and regression
- Two-way ANOVA

Logistic regression

If time: reshaping data

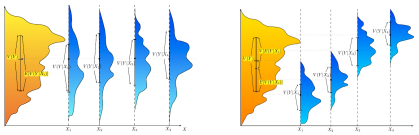
Conclusions

More on ANOVAs

Recall: One-way ANOVA

An Analysis of Variance (ANOVA) is a test that can be used to examine if a set of means are all the same

- H_0: $\mu_1 = \mu_2 = \ldots = \mu_k$
- H_A: $\mu_i \neq \mu_j$ for some i, j



The statistic we use for a one-way ANOVA is the F-statistic

$$F = \frac{\text{between-group variability}}{\text{within-group variability}} = \frac{\frac{1}{K-1}\sum_{i=1}^{K} n_i(\bar{x}_i - \bar{x}_{tot})^2}{\frac{1}{N-K}\sum_{i=1}^{K}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2}$$

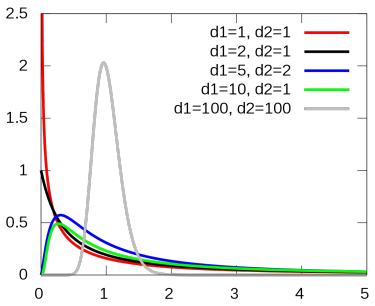
Recall: One-way ANOVA

The F-statistic comes from a F-distribution

- df1 = K - 1
- df2 = N - K

Assumptions underlying a one-way ANOVA

- Data in each group come from normal distributions
- Each group has equal variance (homoskedasticity)



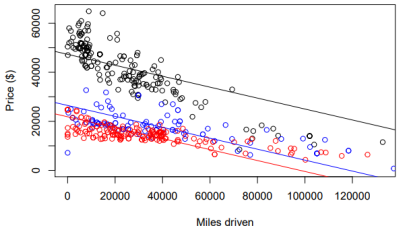
Recall: qualitative predictors

When a qualitative predictor has k levels, we use k -1 dummy variables to code it

- E.g., suppose cars could be a BMW, Mazda, or a Corolla. Then we can code for color using a dummy variable

$$x_{Mazda} = \begin{cases} 1 & if\ \ the\ \ j^{th}\ \ car\ \ is\ \ a\ \ Mazda \\ 0 & if\ \ the\ \ j^{th}\ \ car\ \ is\ \ not\ a\ \ Mazda \end{cases}$$

$$x_{Corolla} = \begin{cases} 1 & if\ \ the\ \ j^{th}\ \ car\ \ is\ \ a\ \ Corolla \\ 0 & if\ \ the\ \ j^{th}\ \ car\ \ is\ \ not\ a\ \ Corolla \end{cases}$$

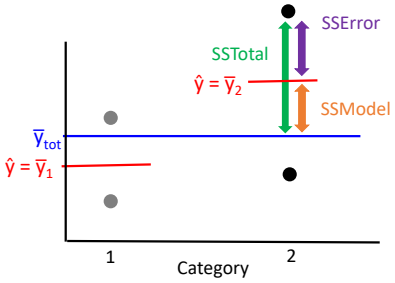


$$y_{Price} = \beta_0 + \beta_1 \cdot x_{Mileage} + \beta_2 \cdot x_{Mazda} + \beta_3 \cdot x_{Corolla} + \epsilon$$

Now we have a model with only qualitative predictors

Hypothesis test based on ANOVA for regression

$$F = \frac{\frac{1}{K-1}\sum_{i=1}^{K} n_i(\bar{y}_i - \bar{y}_{tot})^2}{\frac{1}{N-K}\sum_{i=1}^{K}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i)^2}$$



The ANOVA decomposes the variance as:

- SSTotal = SSModel + SSError

$$y_{ij} - \bar{y}_{tot} = (\hat{y}_{ij} - \bar{y}_{tot}) + (y_{ij} - \hat{y}_{ij})$$

$$(y_{ij} - \bar{y}_{tot})^2 = (\hat{y}_{ij} - \bar{y}_{tot})^2 + (y_{ij} - \hat{y}_{ij})^2$$

$$(y_{ij} - \bar{y}_{tot})^2 = (\bar{y}_i - \bar{y}_{tot})^2 + (y_{ij} - \bar{y}_i)^2$$

$\hat{y}_{ji} = \bar{y}_i$
(the prediction for each class is the group mean)

Let's examine this in R…

Two-way ANOVA

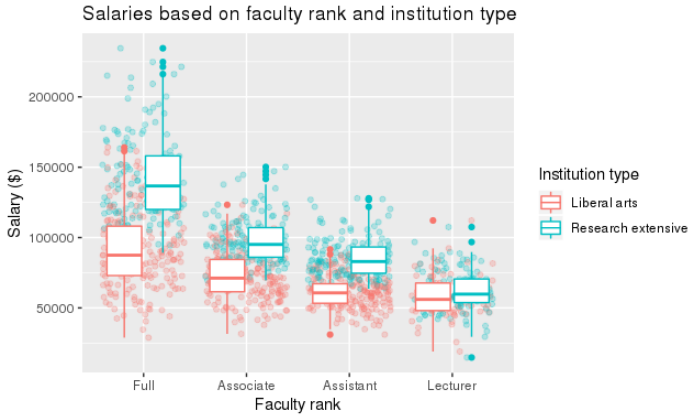
If we have two qualitative variables (and no quantitative variables) then our analysis becomes a two-way ANOVA

Example, how do salary differ for:

- Full professors, Associate professors, Assistant professors and Lecturers
- Extensive research institutions vs. liberal arts colleges

Do salaries seem to be influenced by:

1. The institution type? (Factor A)
2. Faculty rank? (Factor B)



Two-way ANOVA in R

```
> anova_model <- aov(salary_tot ~ Inst_type + rank_name, data = IPED_3)
> summary(anova_model)
             Df    Sum Sq   Mean Sq F value Pr(>F)
Inst_type     1 1.967e+11 1.967e+11   434.7 <2e-16 ***
rank_name     3 5.180e+11 1.727e+11   381.6 <2e-16 ***
Residuals  1255 5.678e+11 4.525e+08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

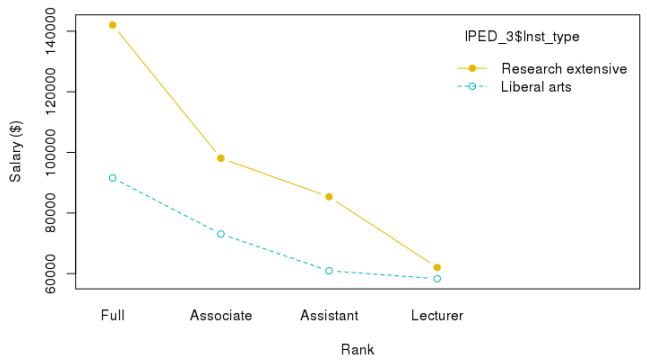
Interactions

We can also examine whether there is an interaction between rank and institution type

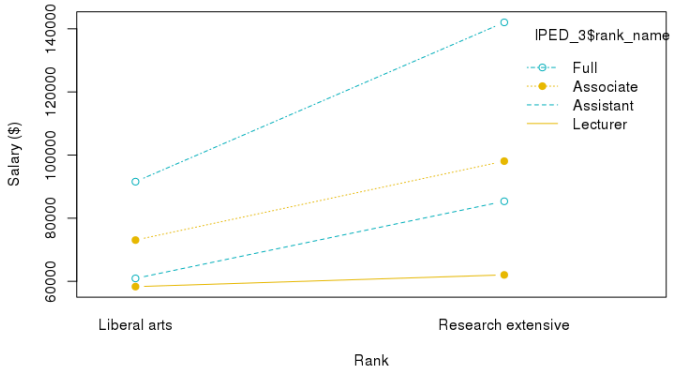
- i.e., does the difference in salaries between faculty ranks differ across institution types?

This similar to using the same slope vs. different slopes model for an interaction between a quantitative and categorical variable

Interaction plots

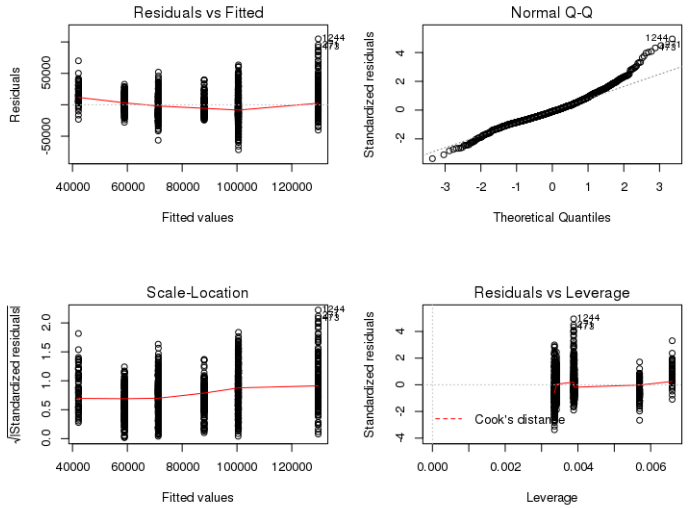


Interaction plots



Two-way ANOVA in R with interaction

```
> fac_anova <- aov(salary_tot ~ Inst_type * rank_name, data = IPED_3)
> summary(fac_anova)
                     Df    Sum Sq   Mean Sq F value Pr(>F)
Inst_type             1 1.967e+11 1.967e+11   496.3 <2e-16 ***
rank_name             3 5.180e+11 1.727e+11   435.7 <2e-16 ***
Inst_type:rank_name   3 7.169e+10 2.390e+10    60.3 <2e-16 ***
Residuals          1252 4.962e+11 3.963e+08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



Testing model assumptions

As we have discussed before, particular assumptions should be met for inferences to be valid when using ANOVAs, including:

- The errors (assessed via the residuals) should be normally distributed
- The variance should be the same across all groups

We can examine these assumptions this through diagnostic plots

Logistic regression

In **logistic regression** we try to predict whether a case belongs to one of two categories (category *a* or category *b*)

- E.g., we could predict if a car is new or used based on price

Making predictions for a categorical variable is called **classification**

- The field of machine learning has developed many classification methods

In logistic regression we build a conditional probability model:

- Pr(Class = a | x)
- Pr(New Car | price = $20,000)

Logistic regression

Question: could we use linear regression to make these predictions?

$Pr(Y = a | x_1) = \beta_0 + \beta_1 x_1$

Problem: we will have negative probabilities and probabilities greater than 1!

Logistic regression

Instead we model the log odds as a linear function of our predictors

$$log(\frac{Pr(Y=a|x)}{Pr(Y=b|x)})$$

log-odds or logit

This scales values in the range of [0 1] to values in the range of (-∞ ∞)

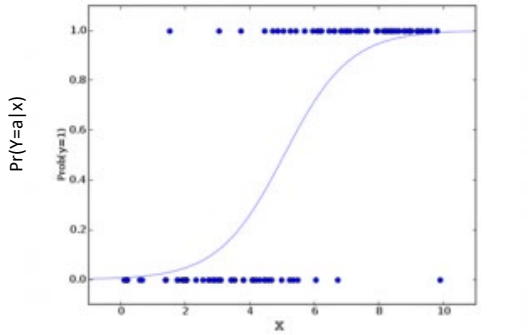
Logistic regression

$$log(\frac{Pr(Y=a|x)}{1-Pr(Y=a|x)}) = \beta_0 + \beta_1 \cdot x$$

Solving for $Pr(Y = a | x)$:

$$Pr(Y = a|x) = \frac{e^{\beta_0+\beta_1 \cdot x_1}}{1+e^{\beta_0+\beta_1 \cdot x_1}}$$

Plotting Pr(Y=a|x) as a function of x

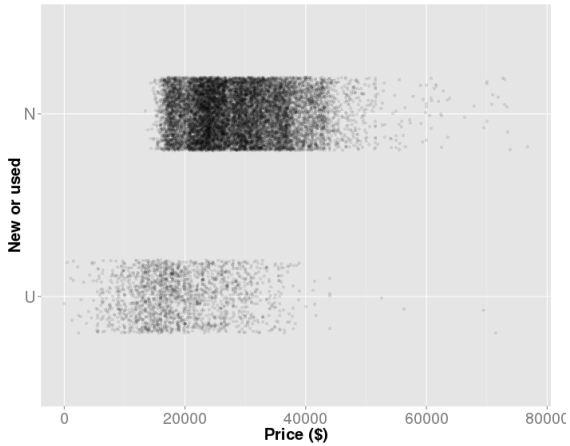


$$Pr(Y = a|x_1) = \frac{e^{\beta_0+\beta_1 \cdot x_1}}{1+e^{\beta_0+\beta_1 \cdot x_1}}$$

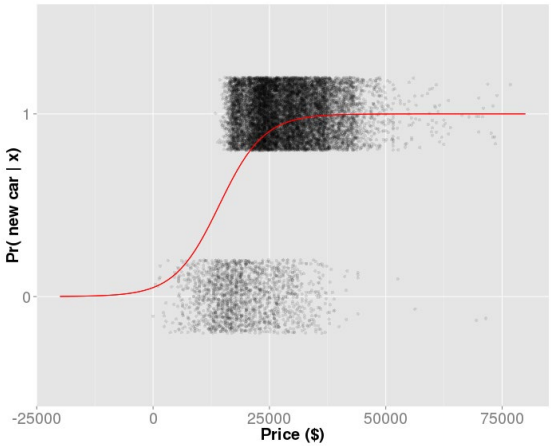
Let’s look at this in R…

Let’s predict whether a Toyota is new or used…

Predicting whether a Toyota sold was new or used based on price



Predicting whether a Toyota sold was new or used based on price



Reshaping data

Wide vs. Narrow data

Plotting data using ggplot requires that data is in the right format (i.e., requires data wrangling)

Often this involves converting data from **wide data** to **narrow data**

Wide data

Person	Age	Weight
Bob	32	128
Alice	24	86
Steve	64	95

Narrow data

Person	Variable	Value
Bob	Age	32
Bob	Weight	128
Alice	Age	24
Alice	Weight	86
Steve	Age	64
Steve	Weight	95

Tidyr:: pivot_longer()

pivot_longer(df, cols) converts data from **wide** to **narrow**

Takes multiple columns and converts them into (***name, value***) pairs

cols specifies which columns to make into the longer format

- The **column names** become **categorical variable levels** in a new variable called **name**
- The **data** in these columns become entries in a variable called **value**

pivot_longer(df, cols = c(Age, Weight))

Wide data

Person	Age	Weight
Bob	32	128
Alice	24	86
Steve	64	95

→

Narrow data

Person	name	value
Bob	Age	32
Bob	Weight	128
Alice	Age	24
Alice	Weight	86
Steve	Age	64
Steve	Weight	95

Tidyr:: pivot_wider()

pivot_wider(df, names_from, values_from) converts data from narrow to wide

- names_from: is the variable who's values will be the new columns
- values_from: is the variable who's values will be in each of the columns

pivot_wider(df, names_from = name, values_from = value)

Narrow data

Person	name	value
Bob	Age	32
Bob	Weight	128
Alice	Age	24
Alice	Weight	86
Steve	Age	64
Steve	Weight	95

→

Wide data

Person	Age	Weight
Bob	32	128
Alice	24	86
Steve	64	95

Let's try it in R…

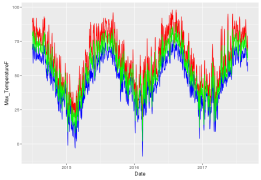
Example of pivot_longer()

Let's use the DataExpo data to plot Max, Min and Mean Temp on the same plot

```
> actual_weather <- read.csv('histWeather.csv')

# convert variables to the appropriate types
actual_weather <- actual_weather %>%
        mutate(Date = as.Date(Date)) %>%
        mutate(PrecipitationIn = as.numeric(as.character(PrecipitationIn)))
```

Example of pivot_longer()



Plot Max, Min and Mean Temp on the same plot

```
> actual_weather %>%
        filter(AirPtCd== "KHVN") %>%
        ggplot(aes(x = Date)) +
        geom_line(aes(y = Max_TemperatureF), col = "red") +
        geom_line(aes(y = Min_TemperatureF), col = "blue") +
        geom_line(aes(y = Mean_TemperatureF), col = "green")
```

Clearly this won't scale well if there are many items to put on the same plot and we can't do facet wrapping with this at all!

Example of pivot_longer()

Let's use pivot_longer() to create long data…

```
actual_weather_long <- actual_weather %>%
        filter(AirPtCd == "KHVN") %>%
        select(-Events) %>%
        pivot_longer(-c(AirPtCd, Date))
```

Exclude these variables but make all others into two long columns (name and value)

Now plot the data comparing Max, Min and mean temperatures

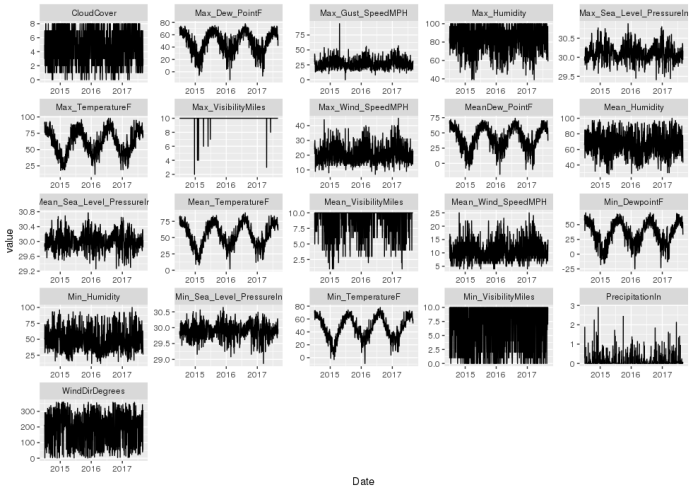
Example of pivot_longer()

```
> actual_weather_long %>%
        filter(name %in% c("Max_TemperatureF",
                "Min_TemperatureF",
                "Mean_TemperatureF")) %>%
        ggplot(aes(x = Date, y = value, color = name)) +
        geom_line()
```

Example of pivot_longer()

Using this approach we can also use facet_wrap to examine plot all the variables in our data frame as a function of the date

```
> actual_weather_long %>%
      ggplot(aes(x = Date, y = value)) +
      geom_line() +
      facet_wrap(~name, scales = "free")
```



Example of pivot_wider()

Suppose we wanted to plot the difference in temperature between New Haven and New York City for each day of the year

- i.e., we want to know if minimum temperature is higher in New York City than New Haven

How could we do this?

- A: ideally we would like to line up the whether of New Haven and New York City on each date and subtract them

```
# Let's make things easier by filter the data to only Max temp in New York City and New Haven
simple_weather <- actual_weather %>%
   select(Date, AirPtCd, Min_TemperatureF) %>%
   filter(AirPtCd %in% c("KHVN", "KNYC"))
```

Example of pivot_wider()

```
# We can then convert the data to a wide format using pivot_wider()
simple_weather_wide <- simple_weather %>%
   pivot_wider(names_from = AirPtCd, values_from = Min_TemperatureF)
```

with these values

Each level will become a new variable (column)

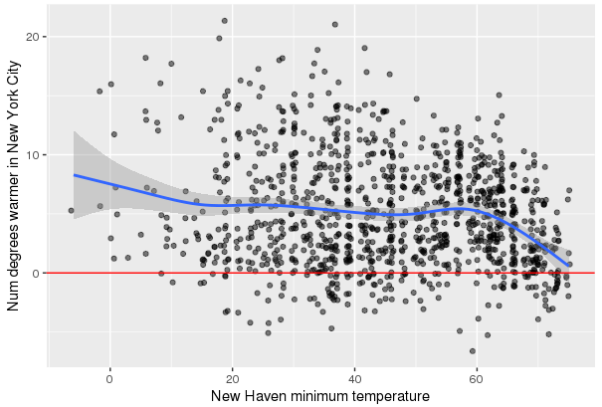
Now take the difference in temperatures between New York City and New Haven and plot the data

Example of pivot_wider()

We can then mutate the data to get the difference in temperature and plot it

```
> simple_weather_wide %>%
    mutate(temp_diff = KNYC - KHVN) %>%
    ggplot(aes( x= Date, y = temp_diff)) +
    geom_point()
```

Example of pivot_wider()

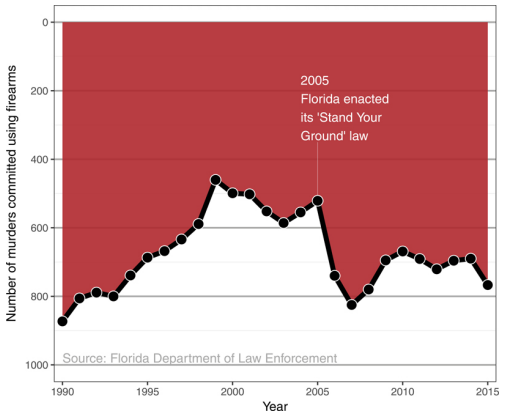


Example of pivot_wider()

Code to create the slightly more elaborate version:

```
> simple_weather_wide %>%
      mutate(temp_diff = KNYC - KHVN) %>%
      ggplot(aes(x = KHVN, y = temp_diff)) +
      geom_jitter(alpha = .5)   +
      xlab("New Haven Temperature") +
      ylab("Num degrees warmer in New York City") +
      geom_smooth() +
      geom_hline(yintercept = 0, col = "red")
```

Ethics



Be ethical!

Conclusions

Course objectives

1. To extend **methods** and **concepts** from intro stats to more complex real world settings

Gain insights on why/how particular methods work
- No math proofs, but we will explore concepts via computational simulations

2. To learn how to analyze and visualize **real data sets** using **the R programming language**

How to find the Truth/trends in a data set and convincingly convey the results to others!

Examples of questions we might look at…

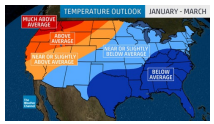
Randomization tests: Is it possible to smell whether someone has Parkinson's disease?

ANOVA: Are all genres of movies equally liked?

Rotten Tomatoes

Data summarization: which airlines have the longest flight delays?

Data wrangling/visualization: How accurate are weather predictions?



Quick Review of central concepts in Intro Statistics

Where is the Truth?

Last question: what was the worst joke of the semester?

Good luck with the end of the semester!

Final exam is on December 14 at 7pm

I will try to hold a review session next week